FACULDADE DE CIÊNCIAS SOCIAIS E HUMANAS
UNIVERSIDADE NOVA DE LISBOA

EXTRACÇÃO DE CLASSES DE OCUPAÇÃO DO SOLO A PARTIR DE IMAGENS DE ALTA RESOLUÇÃO COM RECURSO A ÁRVORES DE DECISÃO

Gonçalo José Marinheiro Revez

Dissertação de Mestrado em Detecção Remota e Sistemas de Informação Geográfica

Março de 2011

Dissertação apresentada para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Detecção Remota e Sistemas de Informação Geográfica, realizada sob a orientação científica de José António Tenedório

Dedicatória pessoal

Agradeço a todos que directa e indirectamente me motivaram para a realização do presente estudo, mas agradeço principalmente à Natália pela compreensão e pelo apoio

AGRADECIMENTOS

Um agradecimento especial ao meu Professor e Orientador José António Tenedório pelo acompanhamento e por ter aceitado fazer parte deste meu desafio.

Aos meus colegas do mestrado, do qual eu tive o enorme prazer de conhecer, o meu obrigado pelos momentos que passamos juntos, e pela partilha da informação e conhecimento.

RESUMO

EXTRACÇÃO DE CLASSES DE OCUPAÇÃO DE SOLO A PARTIR DE IMAGENS DE ALTA RESOLUÇÃO COM RECURSO A ÁRVORES DE DECISÃO

GONÇALO JOSÉ MARINHEIRO REVEZ

PALAVRAS-CHAVE: árvores de decisão, imagens de muito alta resolução, QUEST, C4.5, segmentação

As imagens de satélite e as fotografias aéreas digitais de muito alta resolução são um recurso que hoje em dia é recorrente nos vários estudos e investigações na área da Detecção Remota.

A melhoria das condições tecnológicas de aquisição de imagens digitais é crescente, facto que se tem traduzido no aumento da resolução espacial concomitante com o aumento da resolução espectral como é o caso de WorldView 2. Contudo, a disponibilização crescente de dados com melhores resoluções nem sempre favorece a melhoria dos resultados das classificações automáticas.

Para minimizar esta situação, foram apresentadas várias técnicas na bibliografia, sendo a mais utilizada, a segmentação que procura obter uma classificação ao nível do objecto, em alternativa ao nível do pixel.

Foram realizadas várias investigações no sentido de combinar estes dois tipos de classificação aproveitando as vantagens de cada uma. Todavia, com o aumento da informação ao nível das imagens, os métodos existentes de classificação ao nível do pixel, começaram a ter níveis de performance não adequados para a quantidade de dados a analisar.

As árvores de decisão, sendo um método não paramétrico, ou seja, não recorre à estatística da distribuição dos dados, começou a ter uma maior notoriedade no processo de classificação de imagens, principalmente devido à sua flexibilidade e rapidez de execução.

A presente dissertação procura investigar em maior detalhe algoritmos de geração de árvores de decisão, no contexto da classificação de imagens de muito alta resolução e comparar os algoritmos com os quais se poderão obter melhores resultados.

Desta forma, pretende-se apresentar uma metodologia que consiste em combinar a segmentação, utilizando o software gratuito SPRING, com a classificação supervisionada a partir de algoritmos de geração de árvores de decisão, nomeadamente os algoritmos QUEST e o C4.5.

O algoritmo QUEST é distribuído gratuitamente e é conhecido por colmatar algumas limitações na selecção de atributos de classe para a divisão dos nós que outros algoritmos conhecidos, tais como o CART, apresentam.

A metodologia proposta permite nas várias fases da sua implementação a configuração de parâmetros para as técnicas seleccionadas em cada fase. Assim, foram realizadas várias experiências tendo em conta a combinação de vários parâmetros para comparação e a influência que cada um dos parâmetros tem nos resultados obtidos.

A concordância dos resultados revela que as árvores de decisão são um método eficiente e que retorna melhores resultados face aos métodos existentes que usam a distribuição estatística dos dados. Por outro lado, o algoritmo QUEST que permite várias configurações, demonstra que é um algoritmo igualmente robusto para a geração de árvores de decisão face a outros que são mais conhecidos e que foram alvo de estudo desta dissertação, como o C4.5.

ABSTRACT

SOIL CLASSES EXTRACTION FROM VERY HIGH RESOLUTION IMAGES USING DECISION TREES METHODOLOGIES

GONÇALO JOSÉ MARINHEIRO REVEZ

KEYWORDS: decision tree, very high resolution image, QUEST, C4.5, segmentation

The satellite images and the aerial digital photography of very high resolution are nowadays a resource very used and explored in research and products of Remote Sensing.

The improvement of technological conditions for acquiring digital image is growing, fact that has been seen on the increase of spatial resolution concomitant with the increase of spectral resolution, like the of WorldView2. However, the grown availability of data with better resolutions, not always favours the improvement of automatic classifications results.

The image spatial resolution is a well known grown factor which allows more detail at pixel level. On the other side, the spectral resolution also increases, which turns more difficult to classify the images.

To minimize this situation, several studies were present, where the most known technique used is the segmentation, whose goal is to get a classification based on object instead of pixel.

Other studies were made in order to combine both classifications and to join both advantages of each classification based. However with grow of information at image level, the existence methods of pixel classification, started to have levels of performance not adjusted to the quantity of data to analyse.

The decision trees, being a non parametric method, meaning, do not use statistical distribution of data, started to have a more notoriety in the process of image classification, mostly because of its proprieties like flexibility and speed of execution. The present dissertation has the goal to investigate in more detail, decision tree algorithms in the context of very high resolution image classification and compare the algorithms that are more possible to return better results. It is proposed a methodology, which consists in combining the segmentation, using the software open source SPRING, with the supervised classification, using algorithms to generate decision trees, namely the algorithm QUEST and the C4.5.

The QUEST algorithm is open source and is known to solve some deficiencies on the class attribute selection for the division of node, of other algorithms well known.

The methodology proposed allows in its steps of implementation, the configuration of parameters for the techniques selected in each step. According to this, were made several experiences with the combination of several parameters for comparison and to check the influence of each parameter change has in the results returned.

The precision of the results, reveals that the decision tree are an efficient method and returns better results compared to the existence methods of classification that uses the statistical distribution of data. Another assumption is the fact that the QUEST algorithms with its configurations changes experienced, demonstrates that is a good choice for an algorithm of automatic decision tree generation in terms of image classification, compared with other well known like C4.5 that was investigated in this dissertation.

ÍNDICE

LISTA DE ABREVIATURAS

CART – *Classification and Regression Tree*

CBERS – *China-Brazil Earth Resources Satellite*

CEOS – *Committee of Earth Observations*

CRUISE – *Classification Rule with Unbiased Interaction Selection and Estimation*

EO – *Earth Observation*

ESA – *European Space Agency*

FINGIEE – Fornecimento de Informação Geográfica para Investigação, Ensino e Edição

GEOSS – *Global Earth Observation System of Systems*

GMES – *Global Monitoring for Environment and Security*

IGP – Instituto Geográfico Português

LANDSAT – *Land Satellite*

LCLUC – *Land Cover and Land Use Change*

LUCC – *Land Use and Land Cover Change*

NDVI – *Normalized Difference Vegetation Index*

QA4EO – *Quality Assurance Framework for Earth Observation*

QUEST – *Quick, Unbiased and Efficient Statistical Tree*

SAVI – Adjusted Vegetation Index

TCT – *Tassel Cap Transformation*

ROI – *Region of Interest*

INTRODUÇÃO

Hoje em dia é uma prática comum a utilização de imagens de satélite e de fotografias aéreas digitais de muito alta resolução, na ordem de grandeza dos centímetros por pixel, para os mais variados estudos na área da Detecção Remota.

Com o uso frequente e maior acessibilidade destas imagens, enfrentam-se novos desafios para a investigação de novas metodologias e confrontam-se as metodologias existentes com o objectivo de alcançar resultados com uma maior exactidão.

Face a estas considerações, a presente dissertação procura investigar e relacionar metodologias conhecidas na área da Detecção Remota com outras áreas tecnológicas, nomeadamente Inteligência Artificial através de Árvores de Decisão. Consequentemente, o objectivo consiste em demonstrar que a metodologia das árvores de decisão, constitui uma alternativa favorável e viável na classificação de imagens, face às metodologias existentes e à constante evolução das resoluções espaciais e espectrais.

As Árvores de Decisão são uma técnica muito conhecida na área da Inteligência Artificial e em algoritmos de previsão e qualidade dos dados. Por se considerar que esta metodologia é eficiente, mais intuitiva, simples e robusta, poderá ser uma mais-valia na aplicação de classificação em imagens de muito alta resolução.

Com o aparecimento de mais imagens com muito alta resolução e maior facilidade de acesso, novas técnicas de classificação de imagens vão também surgindo para garantir uma maior exactidão na classificação, como é o caso da segmentação de imagens, referida na bibliografia. Esta técnica tem sido aprofundada e cada vez mais utilizada para imagens com uma grande resolução espacial, demonstrada em vários artigos de investigação.

A presente dissertação não procura fazer comparações directas entre metodologias já conhecidas na temática da Detecção Remota para classificações supervisionadas e não supervisionadas, pois já existem na literatura vários estudos bastante elaborados na comparação entre as várias técnicas.

Relativamente aos principais conceitos aprofundados nesta dissertação, ou seja, árvores de decisão e imagens de muito alta resolução, constata-se que nos últimos cinco anos surgiram novos estudos e novas metodologias que procuram melhorar cada vez

mais a exactidão dos resultados dos problemas presentes. Desta forma, pretende-se acompanhar estes novos estudos, investigar novas soluções com recurso a várias integrações aplicacionais e divulgar novos desafios que procuram encontrar a solução de problemas futuros.

Os algoritmos de geração automática de Árvores de Decisão são um tema conhecido nesta área e aplicado em várias situações concretas. Esta metodologia tem vindo a ganhar maior adesão na detecção remota para a classificação de imagens, principalmente para imagens de muito alta resolução. Os algoritmos mais reconhecidos na comunidade científica e em programas informáticos que procuram ajudar a resolver problemas específicos, são o CART e o C4.5. Contudo, existem outros algoritmos mas que não tiveram tanta adesão. Um dos exemplos é o algoritmo QUEST, que é gratuito e que procura resolver alguns problemas existentes nos algoritmos identificados previamente. O seu código e respectiva aplicação encontram-se também disponíveis para *download*.

O uso do algoritmo QUEST comparativamente com o C4.5 para a geração de árvores de decisão, será o foco de trabalho da presente dissertação, pelas técnicas que apresentam e pelo resultado final pretendido.

Os softwares de código aberto são hoje em dia cada vez mais utilizados tanto pelas comunidades *open-source* como pelas empresas, pois revelam uma alternativa viável e com resultados garantidos e satisfatórios.

A área da Detecção Remota e dos Sistemas de Informação Geográfica tiveram nos últimos anos uma eclosão de aplicações nas mais variadas temáticas. Estas aplicações de código aberto permitem disponibilizar às comunidades uma maior oferta e a possibilidade destas poderem recorrer a aplicações e algoritmos, que previamente eram restritivos. Permitindo explorar e investigar novas ideias e soluções para novos problemas com os dados e com as situações emergentes dos dias de hoje.

Um dos principais objectivos na área da Detecção Remota na vertente do uso de solo é o desenvolvimento de fluxos operacionais de dados de Detecção Remota com a capacidade de disponibilizar e criar aplicações válidas, seguras e avançadas. Estas aplicações, enquadradas no desenvolvimento de sistemas de interpretação de imagens de detecção remota, são prosseguidas por programas internacionais de dados EO, como os programas GEOSS e GMES.

Nestes programas, a sustentabilidade dos projectos desenvolvidos de integração de serviços operacionais, baseados em dados EO de múltiplas fontes, necessitam de processos de validação e harmonização dos dados. Por esta razão surge o programa QA4EO que é responsável pela criação de requisitos que garantem uma plataforma de qualidade para a harmonização e interoperabilidade dos dados EO, dos metadados e das aplicações de informação derivadas (Baraldi, et. al., 2010).

Relativamente à estrutura da presente dissertação, esta está organizada em cinco capítulos.

No primeiro capítulo é elaborado um estado de arte dos principais temas investigados tais como imagens de muito alta resolução, segmentação e árvores de decisão, em que se pretende enquadrar os objectivos e a metodologia proposta para o estudo desenvolvido na presente dissertação.

O segundo capítulo descreve e apresenta os dados que serviram de base de trabalho para a metodologia proposta. Os dados são constituídos por fotografias aéreas de muito alta resolução, adquiridas no âmbito do programa FINGIEE do IGP.

O terceiro capítulo incide com maior detalhe sobre o principal tema investigado, ou seja, as árvores de decisão. Neste capítulo pretende-se também descrever de uma forma sucinta o significado da metodologia das árvores de decisão no contexto da classificação, demonstrar o funcionamento em detalhe dos algoritmos de construção de árvores de decisão com base numa amostra de dados e apresentar os vários programas e algoritmos disponíveis para a construção de árvores de decisão.

Sendo que um dos objectivos da presente dissertação é precisamente a análise comparativa entre dois dos algoritmos enunciados aplicados à classificação de imagens. Ainda neste capítulo são apresentados casos de estudos recentes que demonstram a utilidade das árvores de decisão aplicadas na classificação de imagens.

No quarto capítulo é feita a descrição da metodologia proposta, nomeadamente as suas várias etapas sob a forma de uma arquitectura que descreve o sistema desenvolvido para o problema em estudo na dissertação. Apresentam-se ainda os resultados obtidos através de várias metodologias de teste efectuadas, nomeadamente através de matrizes de confusão para validação dos resultados.

O último e quinto capítulo apresenta todas as conclusões alcançadas, bem como sugestões para novos estudos decorrentes da presente dissertação.

Capitulo I: Enquadramento Geral

I.1 Identificação do Problema e Estado de Arte

O contexto do problema apresentado consiste em investigar e aplicar várias metodologias com árvores de decisão, para a detecção de classes de ocupação de solo em imagens de muito alta resolução.

A evolução das imagens de satélite nos últimos quatro anos muito tem contribuído para a área da Detecção Remota, principalmente desde que o repositório de imagens do satélite Landsat se tornou disponível sem restrições e sem custos à comunidade científica.

Outras iniciativas seguiram o mesmo exemplo, como a parceria entre a China e o Brasil na distribuição de imagens do satélite CBERS e a divulgação da União Europeia para a disponibilização de imagens dos satélites Sentinel-2/-3 com data de lançamento prevista para 2012.

À medida que a resolução espacial das imagens de satélite vai aumentando, aumentam também a qualidade e a quantidade dos dados, o que exige o desenvolvimento de sistemas mais eficientes permitindo analisar quantitativamente e de forma automática os dados das imagens de satélite.

Este objectivo tem sido levado a cabo por diversos programas nas áreas do uso de ocupação do solo, como o LUCC ou o LCLUC. Outros programas têm permitido atingir os mesmos objectivos, como o GEOSS, que procura a harmonização e a interoperabilidade dos dados de observação da terra (EO) gerados a partir de várias fontes e a várias escalas como a global, a regional e a local.

Existe também um programa liderado pela União Europeia em parceria com a ESA, o GMES, cujo objectivo é garantir a sustentabilidade de serviços operacionais de integração dos dados de observação da terra para a monitorização ambiental e a sua segurança.

À medida que a resolução espacial das imagens aumenta, a variância da informação espectral tende também a aumentar, o que prejudica o processo de classificação da imagem, principalmente através dos métodos paramétricos que

recorrem fundamentalmente à resolução espectral da imagem, como é o caso do método de máxima verosimilhança.

Por esta razão, diferentes materiais identificados na classificação aparecem com assinaturas espectrais similares, levando a que a descriminação espectral entre as classes de objectos se torne difícil.

Devido a estas limitações, é expectável que surjam problemas de classificação quando se usa apenas informação espectral. Por exemplo, em ambientes urbanos, existe dificuldade em identificar e distinguir edifícios e estradas porque os seus materiais são espectralmente idênticos. Neste caso, a classificação deste tipo de classes de objecto em imagens de muito alta resolução deve ser acompanhada de mais informação para além da informação espectral. Esta informação adicional pode ajudar a resolver problemas entre classes (Bouziani, et. al., 2010).

Para eliminar este problema, a literatura refere-se a um método de classificação orientado ao objecto, ao contrário dos anteriores que são orientados ao pixel. Ou seja, podem-se usar padrões de segmentos de imagem em vez do pixel. A segmentação de imagem é uma técnica familiar em reconhecimento de padrões (Haralick, et. al., 1985) mas só foi adoptada na última década.

A análise da imagem orientada ao objecto para imagens de muito alta resolução tem demonstrado vantagens significativas. Esta análise subdivide a imagem em regiões homogéneas baseada não só nas propriedades espectrais, mas também na forma, textura, tamanho e outras propriedades topológicas e organiza-as hierarquicamente como objectos de imagens, constituindo segmentos de imagens (Benz et al. 2004).

As metodologias orientadas ao objecto têm sido muito utilizadas com resultados satisfatórios em vários estudos de classificação de uso do solo e de ocupação do solo, como revelam, por exemplo, os trabalhos de Laliberte et al. (2004), Frohn et al. (2005), e de Jensen et al. (2006).

O método de segmentação cria informação adicional que se complementa com a informação espectral. Depois da segmentação, a cada segmento gerado podem-se calcular atributos para serem usados na classificação de imagem. Os atributos usados mais frequentemente são a informação espectral, a textura, a área e o perímetro, tendo sido utilizados em vários estudos, como (Jensen, 2005) e (Shackelford, et. al., 2003). Existem vários métodos de segmentação na literatura e abordados em artigos de

investigação como Carleer, et. al. (2005). A utilização de diversos parâmetros nos vários métodos de segmentação deve ser alvo de vários testes por parte do utilizador, pois estes dependem muito da imagem a analisar e da aplicação a implementar (Bouziani, et. al., 2010).

Os métodos de classificação de imagens de satélite na Detecção Remota sofrem de algumas insuficiências que podem prejudicar a performance operacional dos seus sistemas.

Existem vários factores que podem influenciar essa performance, nomeadamente a precisão da classificação, o tempo de processamento e a memória ocupada dos sistemas, os custos económicos com os dados e com os recursos humanos que necessitam de ter conhecimento específico, a robustez para a mudança das variáveis de entrada e da mudança dos dados, manutenção, escalabilidade e reutilização de acordo com as necessidades dos utilizadores.

Para além destas limitações podem-se referir ainda a morosidade do processo, ou seja, o tempo que decorre desde que se adquire a imagem até à entrega do produto final ao utilizador e o facto de muitos algoritmos científicos serem apresentados na literatura apesar de terem um impacto insignificante nas ferramentas distribuídas por programas comerciais.

Por outro lado o aumento do número de imagens de satélite com melhorias ao nível espacial, espectral e de qualidade temporal superam as capacidades dos actuais sistemas manuais de validação dos dados e dos sistemas de aprendizagem indutiva a partir de classificações supervisionadas de dados de observação da terra.

Desta forma, o custo, a disponibilidade e a qualidade dos dados de referência que por sua vez são derivados de mapas e informação estatística, são actualmente considerados como os factores mais limitativos para a geração e validação de produtos de imagens de Detecção Remota (Baraldi, et. al., 2010).

De modo a optimizar a performance operacional nos sistemas de processamento de dados, devem ser estabelecidas algumas condições aos dados de entrada.

A iniciativa QA4EO, liderada por um grupo de trabalho do CEOS no contexto do programa GMES, veio estabelecer requisitos de qualidade para as unidades de medida das variáveis de entrada, nomeadamente as imagens de satélite. Assim a regra definida para os dados de entrada consiste em estarem radiometricamente calibrados, ou

seja, os números digitais são transformados numa unidade de medida radiométrica de acordo com a iniciativa de garantia de qualidade QA4EO. Desta forma, os dados ficam geometricamente corrigidos, ou seja, projectados num sistema de coordenadas terrestre de referência. Para além disso os dados devem estar validados, isto é, devem disponibilizar informação quantitativa, unívoca e medidas de histórico da qualidade e incerteza geométrica e radiométrica de acordo com as linhas do programa QA4EO.

Relativamente às árvores de decisão, estas começaram a ganhar maior interesse em aplicações na área de detecção remota, demonstrado em vários artigos científicos de autores como (Huang, et. al, 1997), (Muchoney et. al., 2000), (Hodgson et. al., 2003), (Jensen et. al., 2005), muito devido à sua simplicidade e rapidez na previsão de classes para dados exemplo.

Até então, os classificadores a partir de Árvores de Decisão não tinham sido muito explorados pelas comunidades de Detecção Remota para classificações de uso do solo, apesar da sua natureza não paramétrica e as suas propriedades já conhecidas, como simplicidade, flexibilidade e eficiência (Friedl, et. al, 1997).

I.2 Objectivos do Problema

Os objectivos propostos da presente dissertação passam primeiro por fazer um enquadramento do estado da arte na classificação de classes de ocupação do solo para imagens de muito alta resolução e posteriormente pelas metodologias de aprendizagem através das árvores de decisão.

O tema principal é de facto as árvores de decisão, por ser uma metodologia com potencial para resolver muitos dos problemas descritos na literatura na classificação de imagens de muito alta resolução. Deste modo, vão ser explorados dois algoritmos de geração automática das árvores de decisão para posterior classificação de imagens.

Estes algoritmos permitirão a classificação de classes de ocupação do solo para uma área de uma fotografia área digital de resolução espacial de cinquenta centímetros.

A metodologia proposta englobará outras técnicas descritas na literatura que complementam a classificação de imagens, nomeadamente a segmentação. Contudo, esta metodologia não será sujeita a uma análise exaustiva, pois estas já foram realizadas em outros estudos, supra-citados no contexto da presente dissertação.

A metodologia inclui também a integração de várias tecnologias aplicacionais para corresponder aos requisitos propostos, de forma a garantir interoperabilidade entre as diversas aplicações.

Capitulo II: Dados

Os dados foram adquiridos no âmbito de projectos de investigação e ao abrigo do programa FINGIEE[1] do Instituto Geográfico Português. A área de estudo corresponde a um sector da cidade do Montijo. Esta informação é propriedade do IGP e goza da protecção dos direitos de autor, sendo apenas cedido o direito à sua utilização para a finalidade indicada do programa FINGIEE.

II.1 Fotografia Aérea Digital

Os dados foram obtidos pelo método de aquisição de fotografia aérea. Estas foram elaboradas em dois ficheiros distintos, mas orto-rectificados. O Instituto Geográfico Português, nomeadamente a Direcção de Serviços e Geodesia e Cartografia, elaborou a disponibilização destes dados de forma electrónica, conjuntamente com os seus metadados no standard ISO 19115. Estes informam o seguinte:

Tabela 1: Tabela de atributos dos metadados da Fotografia Aérea Digital

Atributos dos metadados	**Valor dos atributos dos metadados**
Sistema de Referencia	EPSG 3763
Titulo	Ortofotocarta IGP 004323B
Título alternativo	004323B
Data	2007-08-01
Abstracto	Folha da série ortofotocartográfica digital do
Objectivo	Destacam-se o suporte a sistemas de
Palavras-chave	Imagem
Resolução espacial	0,5 m
Categoria	imageryBaseMapsEarthCover
Limites geográficos	Oeste: 9.006988111000
Formato de distribuição da imagem	TIFF + World File
Versão do formato da distribuição da imagem	TIFF 6.0
Opções da distribuição da imagem – unidades	Seccionamento de 4 km x 5 km

[1] FINGIEE - http://www.igeo.pt/e-IGEO/precario/FIGIEE.htm

Opções da distribuição da imagem – tamanho	320
Qualidade dos dados	Imagem resultante do mosaico de fotografia

Figura 1: Fotografia aérea digital da uma zona da cidade do Montijo com enquadramento da área de estudo

Capitulo III: Árvores de Decisão

III.1 Visão e Definição

As árvores de decisão são uma metodologia muito comum na área de Inteligência Artificial, principalmente em *Data Mining*, para a determinação de métodos de aprendizagem eficientes no âmbito da classificação e previsão dos dados.

Esta metodologia pode ser utilizada na exploração dos dados em determinadas situações: reduzir um grande volume de dados para uma forma mais compacta, preservando as características essenciais dos dados; classificar os dados em classes de objectos, para que as classes possam ser interpretadas claramente; ou prever valores de variáveis dependentes no futuro a partir de conjuntos de dados constituídos por variáveis independentes e dependentes.

As árvores de decisão são representadas sob forma de estruturas hierárquicas e sequenciais, as quais permitem representar regras sobre um conjunto de dados com o objectivo de criar um método de aprendizagem para classificação ou previsão de novos conjuntos de dados.

No contexto da classificação de imagens, as árvores de decisão não necessitam de nenhum conhecimento ou de nenhuma configuração de parâmetros. É baseada numa aprendizagem supervisionada onde, a partir de um conjunto de dados de treino se pode induzir uma árvore de decisão, do qual são criadas regras sobre os dados para prever classificações de novos conjuntos de dados.

Estas regras estão relacionadas com os atributos de classe e estes podem ser de vários tipos desde binários, nominais ou até valores quantitativos. Por sua vez, as classes têm que ser de tipo qualitativo, ou seja, em categorias ou binário.

De uma forma geral, havendo uma amostra de dados com atributos associados e respectivas classes, a árvore de decisão produz uma sequência de regras numa estrutura hierárquica, em árvore, que pode ser usada para reconhecer as classes.

As árvores de decisão consistem então em tipos de classificação que não recorrem à estatística da distribuição dos dados e foram descritas na bibliografia como tipo de classificações mais precisas (Friedl et. al., 1997 e Xu et al., 2005 e Pal et. al., 2003 e Rogan et al., 2002).

Esta metodologia das árvores de decisão apresenta vantagens em diversas vertentes: é relativamente simples, é explícita e com uma estrutura de classificação intuitiva (Friedl et. al., 1997), tem a capacidade de lidar com relações não lineares entre funcionalidades de classes (Friedl et. al., 1997 e Xu et al., 2005), os dados podem ser representados em escalas de medidas diferentes (Pal et. al., 2003), é rápida em modo de treino (Pal et. al., 2003 e Homer et al., 2004), e rápida em processamento computacional (Pal et al., 2003 e Homer et al., 2004).

As árvores de decisão começaram a ganhar maior interesse em aplicações na área de Detecção Remota, demonstrada em vários artigos científicos dos autores, Huang, et al (1997), Muchoney et al. (2000), Hodgson et al. (2003), Jensen et al. (2005), muito devido à sua simplicidade e rapidez na previsão de classes para dados de treino.

Outros estudos foram realizados, como o de Jensen, et al. (2005) e que obtiveram resultados com sucesso.

Na bibliografia, os classificadores a partir de árvores de decisão, são considerados não métricos (em oposição aos paramétricos e não paramétricos) devido ao uso de heurísticas. Estas heurísticas assumem-se como vantagens sobre as técnicas estatísticas tradicionais, uma vez que não apresentam quaisquer pressupostos sobre a distribuição e independência dos dados (Quinlan 2003 e Jensen, 2005).

III.2 Algoritmos no contexto da classificação

Para a construção de árvores de decisão existem vários algoritmos que permitem criar a árvore de decisão mais eficiente para o problema em análise.

Dos vários algoritmos existentes, todos se baseiam no princípio da divisão por nós do conjunto de dados.

O principal objectivo é dividir o conjunto dos dados em subconjuntos que são mais puros que os dados originais. A pureza dos dados determina-se pela sua homogeneidade em relação aos atributos de classe.

Neste caso, se um conjunto de dados contém apenas um atributo de classe então os dados são homogéneos (puros), por outro lado se contém mais do que um atributo de classe então os dados são heterogéneos.

A técnica mais utilizada para escolher a divisão do conjunto de dados, denomina-se pela técnica "gananciosa". A vantagem desta técnica é ser computacionalmente eficiente independentemente do tamanho do problema.

A técnica gananciosa resume-se em procurar por cada nó o atributo de classe que melhor se ajusta à divisão. Estes atributos são comparados entre si e existem tantos candidatos para a divisão do nó quantos atributos houver.

Alguns algoritmos escolhem o segundo e o terceiro melhor atributo para a divisão do nó e ficam de reserva. Estes atributos são usados quando os conjuntos de dados para o qual se está a construir a árvore de decisão não têm valores atribuídos para o principal atributo seleccionado para a divisão do nó.

Por sua vez, os atributos de classes para a divisão do nó são escolhidos com base numa função de impureza associada ao nó, cujo objectivo é obter o valor mínimo mediante as várias possibilidades de divisão dos vários atributos de classe.

A figura seguinte demonstra um candidato para divisão que vai gerar o nó N1 e o nó N2. A divisão é escolhida entre a diferença do valor da função de impureza do nó N e a soma dos valores da função de impureza dos nós N1 e N2. Aquela que tiver a maior redução na função de impureza é a escolhida.

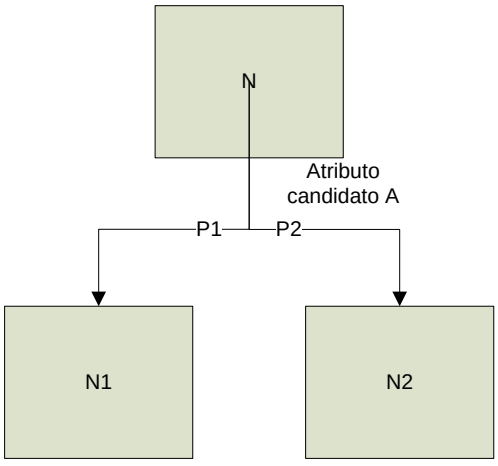


Figura 2: Exemplo de Divisão do Nó por atributo de classe

Assim a fórmula que representa a determinação do melhor atributo de classe para a divisão é representada por:

impur(A,N) = impur(N) – P1.impur(N1) - P2.impur(N2)

Onde *impur* é a função do grau de impureza e P1 e P2 são as proporções do nó N que são distribuídas para o nó N1 e o nó N2.

Existem vários índices para determinar o grau de impureza de um conjunto de dados. Os mais conhecidos são a Entropia, o Índice de Gini e o Erro de Classificação.

$$Entropia = \sum_j -p_j \log_2 p_j$$

$$Indice\ de\ Gini = 1 - \sum_j p_j^2$$

$$Erro\ de\ Classificação = 1 - \max\{p_j\}$$

O Índice de Entropia vai desde o valor zero, para o caso de um conjunto de dados que apenas tenha um atributo de classe, porque o logaritmo de um é zero até ao seu valor máximo que ocorre quando todos os atributos de classe do conjunto de dados têm a mesma probabilidade. A Entropia é, em suma, um conceito que é usado como medida de incerteza sobre um conjunto de dados.

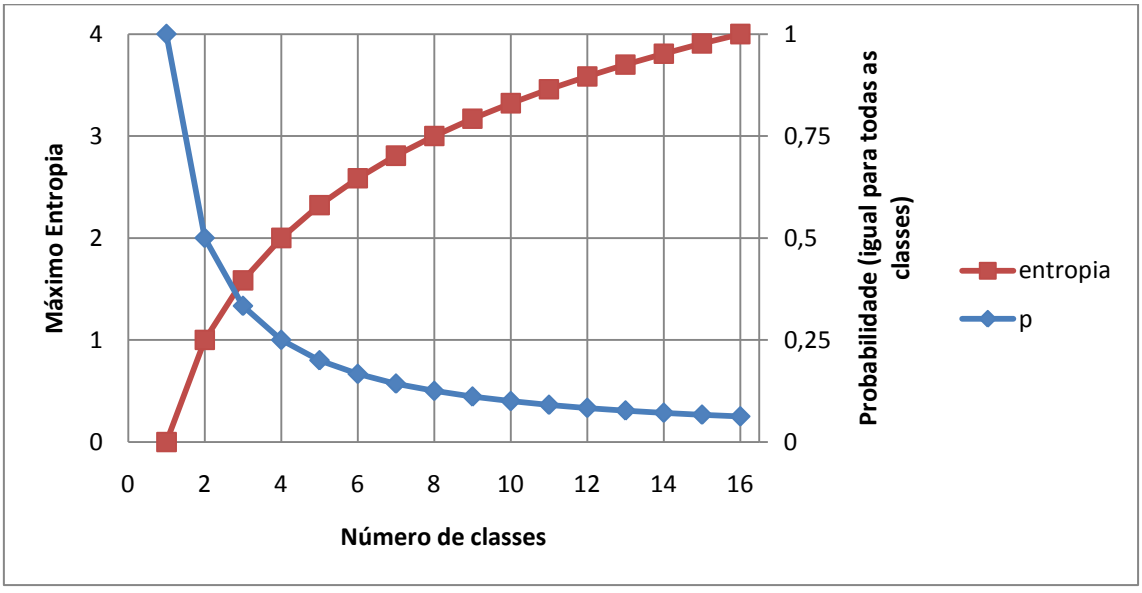


Figura 3: Gráfico de distribuição da medida Entropia e da probabilidade do número de classes

O Índice de Gini para um conjunto de dados puros, ou seja, de apenas um atributo de classe, é zero porque a probabilidade é um, e atinge também o seu valor máximo quando todos os atributos de classe têm a mesma probabilidade.

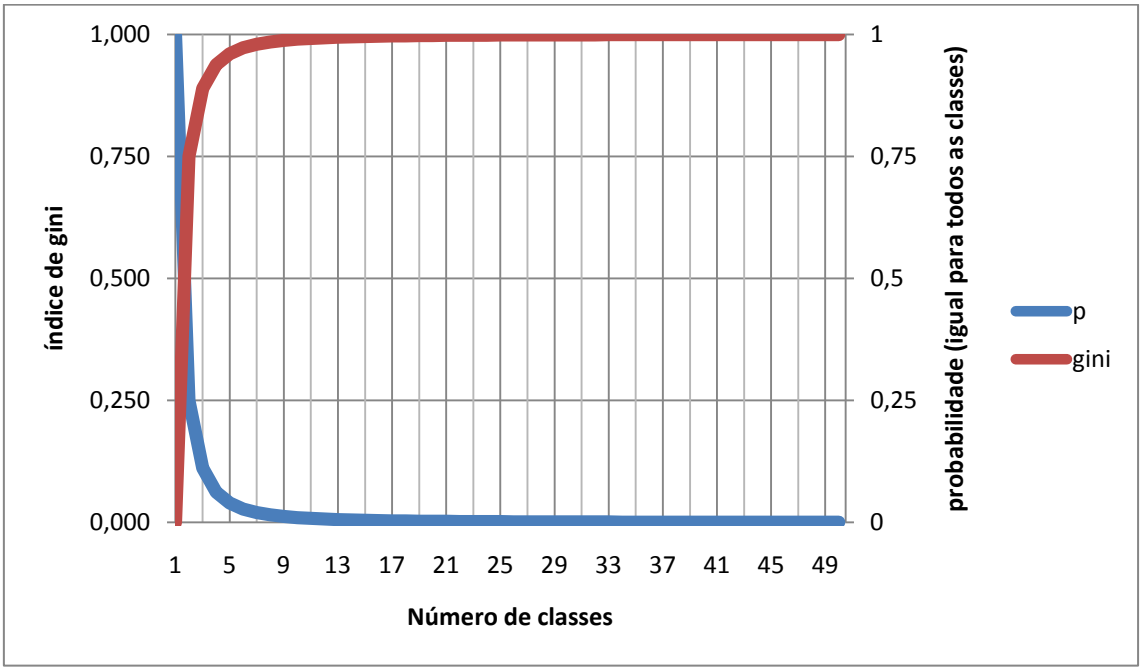


Figura 4: Gráfico de distribuição do índice de Gini e da probabilidade do número de classes

O índice do Erro de Classificação é semelhante aos índices referidos anteriormente quando o conjunto de dados é puro, ou seja, também tem o valor zero. O máximo do índice do Erro de Classificação é sempre igual ao máximo do índice de Gini.

Os algoritmos mais utilizados são o ID3, o C4.5, e C5.0 e o CART (classificação e árvores de regressão). Na globalidade, os algoritmos de árvores de decisão são recursivos.

III.3 O funcionamento da árvore de decisão

O objectivo na construção da árvore de decisão é obter uma árvore com os elementos nó folha mais puros, o que garante uma maior precisão na classificação.

O funcionamento da árvore de decisão parte de um conjunto de dados que contém atributos de classes que definem os dados e as classes associadas.

Para determinar a divisão dos nós da árvore usa-se uma função com um dos índices descritos anteriormente para medir a pureza do conjunto de dados.

De seguida apresenta-se um exemplo demonstrativo de uma área de treino do conjunto de dados aplicado à metodologia da presente dissertação para descrever os vários passos na construção da árvore de decisão.

A função utilizada para determinar o grau de impureza é a Entropia e a amostra é constituída por seis classes: regadio, solo a descoberto, floresta, casas, estradas e mato. Os atributos de classes são respectivamente, o NDVI, as bandas B1, B2, B3 e B4 (Infra-vermelho próximo) e por fim o atributo segmentação.

Tabela 2: Amostra de área de treino do conjunto de dados

Classes Solo	NDVI	B1	B2	B3	B4	Segmentação
regadio	152	79	112	112	164	297
regadio	158	77	109	108	161	297
regadio	145	80	113	113	162	297
regadio	143	78	111	110	157	297
soloadescoberto	72	178	168	147	170	9
soloadescoberto	70	177	167	146	168	9
soloadescoberto	76	176	167	146	171	175
floresta	200	76	102	92	156	337
floresta	191	79	104	98	162	337
floresta	177	80	104	86	136	764
floresta	196	73	94	77	129	764
floresta	212	70	90	72	127	764
floresta	189	77	99	84	138	764
casas	168	204	159	111	171	1038
casas	134	212	168	131	182	1034
casas	119	203	165	131	174	1061
casas	146	191	153	116	167	1069
estradas	10	158	155	144	139	963
estradas	13	159	159	148	144	816
estradas	17	175	171	156	154	991
mato	169	84	82	72	111	416
mato	116	132	116	98	129	525
mato	108	123	111	95	122	710
mato	120	114	101	87	116	718

Começa-se por calcular a Entropia da tabela de dados apresentada, e esta é calculada da seguida forma:

Entropia da Tabela de Dados = -4/10 * log2(4/10) + (-3/10 * log2(3/10)) + (-6/10 * log2(6/10)) + (-4/10 * log2(4/10)) + (-3/10 * log2(3/10)) + (-4/10 * log2(4/10)) = **2,542481**

Para comparar a pureza dos dados recorre-se a uma medida denominada de Ganho de Informação, que permite determinar o ganho por cada divisão do conjunto de dados baseado nos atributos de classes.

A fórmula é definida por:

Ganho de Informação (i) = Entropia da tabela pai – Soma (k / n * Entropia de cada valor k do subconjunto da tabela)

Usando um atributo de classe seleccionado, demonstra-se o ganho de informação obtido pela divisão do conjunto de dados original no subconjunto de dados agrupados associado ao atributo de classe e são calculados os graus de impureza de cada um através da função de Entropia.

De seguida apresentam-se todos os atributos de classe candidatos com o respectivo cálculo do ganho de informação para determinar a escolha do melhor atributo de classe.

Classes Solo	B1
floresta	70
floresta	73
floresta	76
regadio	77
floresta	77
regadio	78
regadio	79
floresta	79
regadio	80
floresta	80
mato	84
mato	114
mato	123
mato	132
estradas	158
estradas	159
estradas	175

Classes Solo	B1
regadio	77
floresta	77
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	B1
regadio	79
floresta	79
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	B1
regadio	80
floresta	80
Entropia	1

soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
casas	191
casas	203
casas	204
casas	212

Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B1 é calculado da seguinte forma:

Ganho Informação (B1) = 2,542481 - (2/24 * 1 + 2/24 * 1 + 2/24 * 1) = **2,292481**

O atributo de classe NDVI não tem valores repetidos de forma a agrupá-los em subconjuntos, como tal, o valor de Ganho de Informação é o mesmo da tabela inicial.

Classes Solo	B2
mato	82
floresta	90
floresta	94
floresta	99
mato	101
floresta	102
floresta	104
floresta	104
regadio	109
regadio	111
mato	111
regadio	112
regadio	113
mato	116
casas	153
estradas	155
estradas	159
casas	159
casas	165
soloadescoberto	167
soloadescoberto	167

Classes Solo	B2
floresta	104
floresta	104

Entropia	0
Gini	1
Erro Classificação	1

Classes Solo	B2
regadio	111
mato	111

Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	B2
estradas	159
casas	159

Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	B2

soloadescoberto	168
casas	168
estradas	171

soloadescoberto	167
soloadescoberto	167
Entropia	0
Gini	1
Erro Classificação	1

Classes Solo	**B2**
soloadescoberto	168
casas	168
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B2 é calculado da seguinte forma:

Ganho Informação (B2) = 2,542481 - (2/24 * 0 + 2/24 * 1 + 2/24 * 1 + 2/24 * 0 + 2/24 * 1) = **2,292481**

Classes Solo	**B3**
mato	72
floresta	72
floresta	77
floresta	84
floresta	86
mato	87
floresta	92
mato	95
floresta	98
mato	98
regadio	108
regadio	110
casas	111
regadio	112
regadio	113
casas	116
casas	131
casas	131

Classes Solo	**B3**
mato	72
floresta	72
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	**B3**
floresta	98
mato	98
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	**B3**
casas	131
casas	131
Entropia	0
Gini	1

estradas	144
soloadescoberto	146
soloadescoberto	146
soloadescoberto	147
estradas	148
estradas	156

Erro Classificação	1

Classes Solo	**B3**
soloadescoberto	146
soloadescoberto	146
Entropia	0
Gini	1
Erro Classificação	1

O ganho de informação para o atributo de classe B3 é calculado da seguinte forma:

Ganho Informação (B3) = =2,542481 - (2/24 * 1 + 2/24 * 1 + 2/24 * 0 + 2/24 * 0) = **2,375814333**

Classes Solo	**B4**
mato	111
mato	116
mato	122
floresta	127
floresta	129
mato	129
floresta	136
floresta	138
estradas	139
estradas	144
estradas	154
floresta	156
regadio	157
regadio	161
floresta	162
regadio	162
regadio	164
casas	167
soloadescoberto	168
soloadescoberto	170
casas	171
soloadescoberto	171
casas	174
casas	182

Classes Solo	**B4**
floresta	129
mato	129
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	**B4**
floresta	162
regadio	162
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	**B4**
casas	171
soloadescoberto	171
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B4 é calculado da seguinte forma:

Ganho Informação (B4) = 2,542481 - (2/24 * 1 + 2/24 * 1 + 2/24 * 1) = **2,292481**

Classes Solo	Segmentação
soloadescoberto	9
soloadescoberto	9
soloadescoberto	175
regadio	297
regadio	297
regadio	297
regadio	297
floresta	337
floresta	337
mato	416
mato	525
mato	710
mato	718
floresta	764
floresta	764
floresta	764
floresta	764
estradas	816
estradas	963
estradas	991
casas	1034
casas	1038
casas	1061
casas	1069

Classes Solo	Segmentação
soloadescoberto	9
soloadescoberto	9
Entropia	0
Gini	1
Erro Classificação	1

Classes Solo	Segmentação
regadio	297
regadio	297
regadio	297
regadio	297
Entropia	0
Gini	1
Erro Classificação	1

Classes Solo	Segmentação
floresta	337
floresta	337
Entropia	0
Gini	1
Erro Classificação	1

Classes Solo	Segmentação
floresta	764
floresta	764
floresta	764
floresta	764
Entropia	0
Gini	1
Erro Classificação	1

O ganho de informação para o atributo de classe Segmentação é calculado da seguinte forma:

Ganho Informação (Segmentação) = 2,542481 - (2/24 * 1+ 4/24 * 0 + 2/24 * 0 + 4/24 * 0) = **2,459147667**

Os resultados obtidos dos ganhos de informação por cada atributo de classe são demonstrados na seguinte tabela:

Ganho	NDVI	B1	B2	B3	B4	Segmentação
Entropia	2,542481	2,292481	2,292481	2,375814	2,292481	2,459148

Uma vez que os valores do Ganho de Informação são iguais nos atributos de classe B1, B2 e B4, procede-se à técnica gananciosa para escolher o atributo de classe mais adequado.

Para o atributo de classe B1, caso seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos.

Classes Solo	B1
floresta	70
floresta	73
floresta	76
regadio	77
floresta	77
regadio	78
regadio	79
floresta	79
regadio	80
floresta	80
mato	84
mato	114
mato	123
mato	132
estradas	158
estradas	159
estradas	175

Classes Solo	B1
floresta	70
floresta	73
floresta	76
regadio	77
floresta	77
regadio	78
regadio	79
floresta	79
regadio	80
floresta	80

Entropia 0,970951

Classes Solo	B1
mato	84
mato	114
mato	123
mato	132

soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
casas	191
casas	203
casas	204
casas	212
Entropia	2,542481

estradas	158
estradas	159
estradas	175
soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
casas	191
casas	203
casas	204
casas	212
Entropia	1,985228

O grau de impureza obtido pela divisão da tabela e por conseguinte do nó, pelo atributo B1 é obtido por:

Impureza (B1) = 2,54248125036058 – (0,970951 - 1,985228) = 3,556759

Para o atributo de classe B2, se seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos.

Classes Solo	**B2**
mato	82
floresta	90
floresta	94
floresta	99
mato	101
floresta	102
floresta	104
floresta	104
regadio	109
regadio	111
mato	111
regadio	112
regadio	113
mato	116
casas	153
estradas	155
estradas	159

Classes Solo	**B2**
mato	82
floresta	90
floresta	94
floresta	99
mato	101
floresta	102
floresta	104
floresta	104
regadio	109
regadio	111
mato	111
regadio	112
regadio	113
mato	116
Entropia	1,556657

Classes Solo	**B2**

casas	159
casas	165
soloadescoberto	167
soloadescoberto	167
soloadescoberto	168
casas	168
estradas	171
Entropia	2,542481

casas	153
estradas	155
estradas	159
casas	159
casas	165
soloadescoberto	167
soloadescoberto	167
soloadescoberto	168
casas	168
estradas	171
Entropia	1,570951

O grau de impureza obtido pela divisão da tabela e por conseguinte do nó, pelo atributo B2 é obtido por:

Impureza (B2) = 2,54248125036058 – (1,556657 - 1,570951) = 2,556775

Para o atributo de classe B4, se seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos.

Classes Solo	B4
mato	111
mato	116
mato	122
floresta	127
floresta	129
mato	129
floresta	136
floresta	138
estradas	139
estradas	144
estradas	154
floresta	156
regadio	157
regadio	161
floresta	162
regadio	162
regadio	164

Classes Solo	B4
mato	111
mato	116
mato	122
floresta	127
floresta	129
mato	129
floresta	136
floresta	138
estradas	139
estradas	144
estradas	154
floresta	156
regadio	157
regadio	161
floresta	162
regadio	162
regadio	164

casas	167
soloadescoberto	168
soloadescoberto	170
casas	171
soloadescoberto	171
casas	174
casas	182
Entropia	2,542481

Entropia 1,923559

Classes Solo	**B4**
casas	167
soloadescoberto	168
soloadescoberto	170
casas	171
soloadescoberto	171
casas	174
casas	182
Entropia	0,985228

O grau de impureza obtido pela divisão da tabela e por conseguinte do nó, pelo atributo B4 é obtido por:

Impureza (B4) = 2,54248125036058 – (1,923559 - 0,985228) = 3,480812

Comparando os valores de impureza de ambos os atributos candidatos para a divisão do nó, obtém-se um valor mais baixo do grau de impureza para o atributo de classe B2, pelo que será esse o candidato seleccionado visto garantir melhor ganho de informação. Assim, este passa a ser o primeiro nó na árvore de decisão, ou seja, o nó raiz.

Depois de identificado o atributo de classe óptimo, divide-se o conjunto de dados original baseado nesse atributo. Para este subconjunto não necessitamos do atributo de classe previamente calculado porque é redundante.

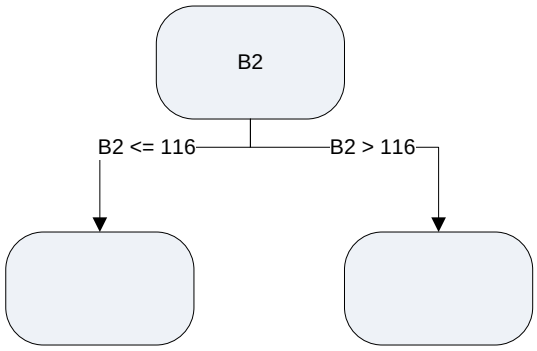
A divisão da tabela é feita mediante duas condições. A primeira consiste nos elementos em que B2<=116 e a segunda consiste nos elementos da tabela em que B2 > 116, demonstrada nas tabelas de dados seguintes.

Classes Solo	**NDVI**	**B1**	**B3**	**B4**	**Segmentação**
mato	169	84	72	111	416
floresta	212	70	72	127	764
floresta	196	73	77	129	764
floresta	189	77	84	138	764
mato	120	114	87	116	718

floresta	200	76	92	156	337
floresta	191	79	98	162	337
floresta	177	80	86	136	764
regadio	158	77	108	161	297
regadio	143	78	110	157	297
mato	108	123	95	122	710
regadio	152	79	112	164	297
regadio	145	80	113	162	297
mato	116	132	98	129	525

Classes Solo	**NDVI**	**B1**	**B3**	**B4**	**Segmentação**
casas	146	191	116	167	1069
estradas	10	158	144	139	963
estradas	13	159	148	144	816
casas	168	204	111	171	1038
casas	119	203	131	174	1061
soloadescoberto	76	176	146	171	175
soloadescoberto	70	177	146	168	9
soloadescoberto	72	178	147	170	9
casas	134	212	131	182	1034
estradas	17	175	156	154	991

A árvore de decisão é provisoriamente construída da seguinte forma:



De seguida procede-se a uma segunda iteração de cada uma das tabelas para determinar o atributo de classe para dividir o nó.

Classes Solo	B1
floresta	70
floresta	73
floresta	76
floresta	77
regadio	77
regadio	78
floresta	79
regadio	79
floresta	80
regadio	80
mato	84
mato	114
mato	123
mato	132

Entropia 1,570951

Classes Solo	B1
regadio	77
floresta	77

Entropia 1
Gini 0,5
Erro Classificação 0,5

Classes Solo	B1
regadio	79
floresta	79

Entropia 1
Gini 0,5
Erro Classificação 0,5

Classes Solo	B1
regadio	80
floresta	80

Entropia 1
Gini 0,5
Erro Classificação 0,5

O ganho de informação para o atributo de classe B1 é calculado da seguinte forma:

Ganho Informação (B1) = 1,570951 - (2/14 * 1 + 2/14 * 1 + 2/14 * 1) = **1,142379**

Classes Solo	B3
floresta	72
mato	72
floresta	77
floresta	84
floresta	86
mato	87
floresta	92
mato	95

Classes Solo	B3
mato	72
floresta	72

Entropia 1
Gini 0,5
Erro Classificação 0,5

Classes Solo	B3
floresta	98

floresta	98
mato	98
regadio	108
regadio	110
regadio	112
regadio	113
Entropia	1,570951

mato	98
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B3 é calculado da seguinte forma:

Ganho Informação (B3) = 1,570951 - (2/14 * 1 + 2/14 * 1)= **1,285236**

Classes Solo	**B4**
mato	111
mato	116
mato	122
floresta	127
floresta	129
mato	129
floresta	136
floresta	138
floresta	156
regadio	157
regadio	161
floresta	162
regadio	162
regadio	164
Entropia	1,570951

Classes Solo	**B4**
floresta	129
mato	129
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	**B4**
floresta	162
regadio	162
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B4 é calculado da seguinte forma:

Ganho Informação (B4) = 1,570951 - (2/14 * 1 + 2/14 * 1) = **1,285236**

Classes	**Segmentação**

Classes Solo	**Segmentação**

Solo	
regadio	297
regadio	297
regadio	297
regadio	297
floresta	337
floresta	337
mato	416
mato	525
mato	710
mato	718
floresta	764
floresta	764
floresta	764
floresta	764

Entropia 1,570950594

regadio	297
regadio	297
regadio	297
regadio	297

Entropia 0

Gini 1

Erro Classificação 1

Classes Solo	Segmentação
floresta	337
floresta	337

Entropia 0

Gini 1

Erro Classificação 1

Classes Solo	Segmentação
floresta	764
floresta	764
floresta	764
floresta	764

Entropia 0

Gini 1

Erro Classificação 1

O ganho de informação para o atributo de classe Segmentação é calculado da seguinte forma:

Ganho Informação (Seg) = 1,570951 - (4/14 * 0 + 2/14 * 0 + 4/14 * 0) = **1,570951**

Os resultados obtidos são demonstrados na seguinte tabela:

Ganho	NDVI	B1	B3	B4	Segmentação
Entropia	-	1,142379	1,285236	1,285236	1,570951

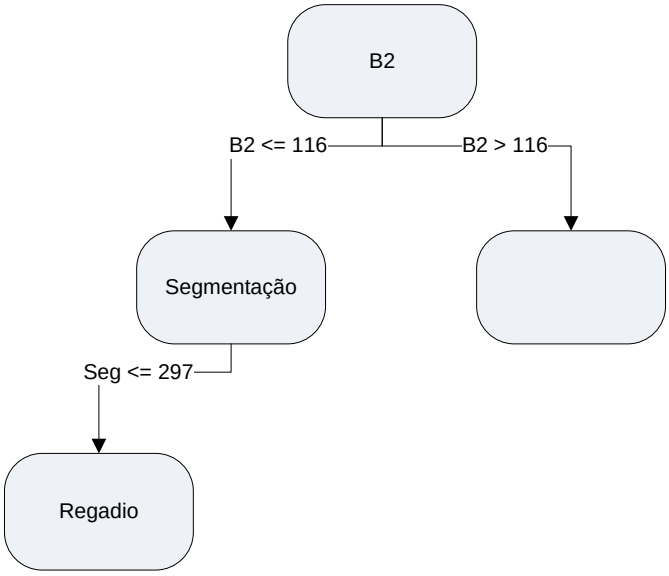
O atributo de classe Segmentação é o que apresenta o maior ganho de informação, por isso é o escolhido.

Classes Solo	NDVI	B1	B3	B4
regadio	143	78	110	157
regadio	158	77	108	161
regadio	145	80	113	162
regadio	152	79	112	164

Classes Solo	NDVI	B1	B3	B4
floresta	200	76	92	156
floresta	191	79	98	162
mato	169	84	72	111
mato	116	132	98	129
mato	108	123	95	122
mato	120	114	87	116
floresta	212	70	72	127
floresta	196	73	77	129
floresta	177	80	86	136
floresta	189	77	84	138

Após a divisão do conjunto de dados pelo atributo de classe segmentação, verifica-se que o primeiro subconjunto está associado a uma classe única e pura, o regadio, e consequentemente esta é associada ao elemento nó folha da árvore. Apenas o último subconjunto dos dados necessita de ser dividido e proceder à verificação de que o grau de pureza dos dados é melhor nos seus subconjuntos.

Assim a árvore é representada da seguinte forma:



Mais uma vez para este subconjunto não necessitamos do atributo de classe previamente calculado porque é redundante.

A próxima iteração consiste no caso em que o atributo de classe Segmentação é superior a 297. De seguida apresentam-se as divisões por cada atributo de classe candidatas.

Classes Solo	NDVI
mato	108
mato	116
mato	120
mato	169
floresta	177
floresta	189
floresta	191
floresta	196
floresta	200
floresta	212
Entropia	0,970951

Classes Solo	B1
floresta	70
floresta	73
floresta	76
floresta	77
floresta	79
floresta	80
mato	84
mato	114
mato	123
mato	132
Entropia	0,970951

Estes dois atributos como não têm valores que podem ser agregados, o valor do Ganho da Informação é igual ao da Entropia.

Classes Solo	B3
floresta	72
mato	72
floresta	77
floresta	84
floresta	86
mato	87
floresta	92
mato	95
floresta	98
mato	98
Entropia	0,970951

Classes Solo	B3
floresta	72
mato	72
Entropia	1
Gini	0,5
Erro Classificação	0,5

Classes Solo	B3
floresta	98
mato	98
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B3 é calculado da seguinte forma:

Ganho Informação (B3) = 0,970951 - (2/10 * 1 + 2/10 * 1) = **0,570951**

Classes Solo	B4
mato	111
mato	116
mato	122
floresta	127
floresta	129
mato	129
floresta	136
floresta	138
floresta	156
floresta	162
Entropia	0,970951

Classes Solo	B4
floresta	129
mato	129
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B4 é calculado da seguinte forma:

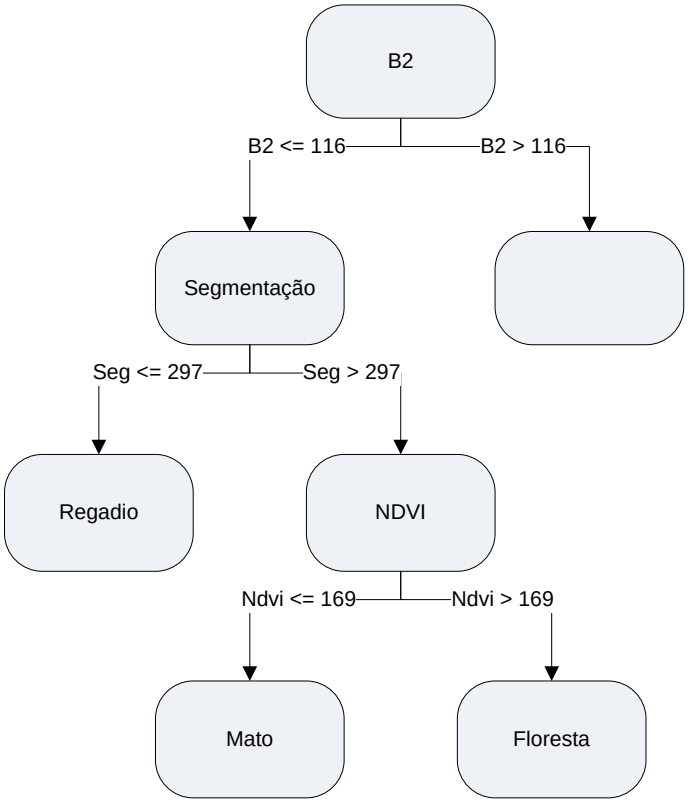
Ganho Informação (B4) = 0,970951 - (2/10 * 1) = **0,770951**

Os resultados obtidos dos ganhos de informação por cada atributo de classe são demonstrados na seguinte tabela:

Ganho	NDVI	B1	B3	B4
Entropia	0,970951	0,970951	0,570951	0,770951

Tanto o atributo de classe NDVI como o B1 podem ser atributos candidatos pois apresentam o maior ganho de informação. Uma vez que ambos podem ser divididos em duas classes puras, ambos podem ser seleccionados. Neste caso escolhe-se o primeiro, ou seja, o atributo de classe NDVI.

A representação da árvore de decisão é demonstrada da seguinte forma:



Procede-se agora ao cálculo de selecção do melhor atributo para o nó à direita do nó raiz.

Classes Solo	NDVI
estradas	10
estradas	13
estradas	17
soloadescoberto	70
soloadescoberto	72
soloadescoberto	76
casas	119
casas	134
casas	146
casas	168
Entropia	1,570951

Classes Solo	B1
estradas	158
estradas	159
estradas	175
soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
casas	191
casas	203
casas	204
casas	212
Entropia	1,570951

Visto estes dois atributos não terem valores que possam ser agregados, o valor do Ganho da Informação é igual ao da Entropia.

Classes Solo	B3
casas	111
casas	116
casas	131
casas	131
estradas	144
soloadescoberto	146
soloadescoberto	146
soloadescoberto	147
estradas	148
estradas	156
Entropia	1,570951

Classes Solo	B3
casas	131
casas	131
Entropia	0
Gini	1
Erro Classificação	1

Classes Solo	B3
soloadescoberto	146
soloadescoberto	146
Entropia	0
Gini	1
Erro Classificação	1

O ganho de informação para o atributo de classe B3 é calculado da seguinte forma:

Ganho Informação (B3) = 1,570951- (2/10 * 0 + 2/10 * 0) = **1,570951**

Classes Solo	B4
estradas	139

Classes Solo	B4
casas	171

estradas	144
estradas	154
casas	167
soloadescoberto	168
soloadescoberto	170
casas	171
soloadescoberto	171
casas	174
casas	182
Entropia	1,570951

soloadescoberto	171
Entropia	1
Gini	0,5
Erro Classificação	0,5

O ganho de informação para o atributo de classe B4 é calculado da seguinte forma:

Ganho Informação (B4) = 1,570951 - (2/10 * 1) = **1,370951**

Classes Solo	**Segmentação**
soloadescoberto	9
soloadescoberto	9
soloadescoberto	175
estradas	816
estradas	963
estradas	991
casas	1034
casas	1038
casas	1061
casas	1069
Entropia	1,570950594

Classes Solo	**Segmentação**
soloadescoberto	9
soloadescoberto	9
Entropia	0
Gini	1
Erro Classificação	1

O ganho de informação para o atributo de classe Segmentação é calculado da seguinte forma:

Ganho Informação (Segmentação) = 1,570951 - (2/10 * 0) = **1,570951**

Os resultados obtidos do ganho de informação de cada atributo de classe são demonstrados na seguinte tabela:

Ganho	NDVI	B1	B3	B4	Segmentação
Entropia	1,570951	1,570951	1,570951	1,370951	1,570951

Uma vez que os valores do Ganho de Informação são iguais nos atributos de classe NDVI, B1, B3 e Segmentação, procede-se à técnica gananciosa para escolher o atributo de classe mais adequado.

Para o atributo de classe NDVI, caso seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos:

Classes Solo	NDVI
estradas	10
estradas	13
estradas	17
soloadescoberto	70
soloadescoberto	72
soloadescoberto	76
casas	119
casas	134
casas	146
casas	168

Entropia 1,570950594

Classes Solo	NDVI
estradas	10
estradas	13
estradas	17

Entropia 0

Classes Solo	NDVI
soloadescoberto	70
soloadescoberto	72
soloadescoberto	76
casas	119
casas	134
casas	146
casas	168

Entropia 0,985228136

Impureza 2,55617873

racio ganho 1,594504396

Para o atributo de classe B1, caso seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos:

Classes Solo	B1
estradas	158

Classes Solo	B1
estradas	158

estradas	159
estradas	175
soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
casas	191
casas	203
casas	204
casas	212
Entropia	1,570950594

estradas	159
estradas	175
Entropia	0

Classes Solo	**B1**
soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
casas	191
casas	203
casas	204
casas	212
Entropia	0,985228136

Impureza	2,55617873
racio ganho	1,594504396

Para o atributo de classe B3, caso seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos:

Classes Solo	**B3**
casas	111
casas	116
casas	131
casas	131
estradas	144
soloadescoberto	146
soloadescoberto	146
soloadescoberto	147
estradas	148
estradas	156
Entropia	1,570950594

Classes Solo	**B3**
casas	111
casas	116
casas	131
casas	131
Entropia	0

Classes Solo	**B3**
estradas	144
soloadescoberto	146
soloadescoberto	146
soloadescoberto	147
estradas	148
estradas	156
Entropia	1

Impureza	2,570950594
racio ganho	1,570950594

Para o atributo de classe Segmentação, caso seleccionado, temos a seguinte divisão da tabela de dados, de forma a separar melhor as classes de atributos:

Classes Solo	Segmentação
soloadescoberto	9
soloadescoberto	9
soloadescoberto	175
estradas	816
estradas	963
estradas	991
casas	1034
casas	1038
casas	1061
casas	1069

Entropia 1,570950594

Classes Solo	Segmentação
soloadescoberto	9
soloadescoberto	9
soloadescoberto	175

Entropia 0

Classes Solo	Segmentação
estradas	816
estradas	963
estradas	991
casas	1034
casas	1038
casas	1061
casas	1069

Entropia 0,985228136

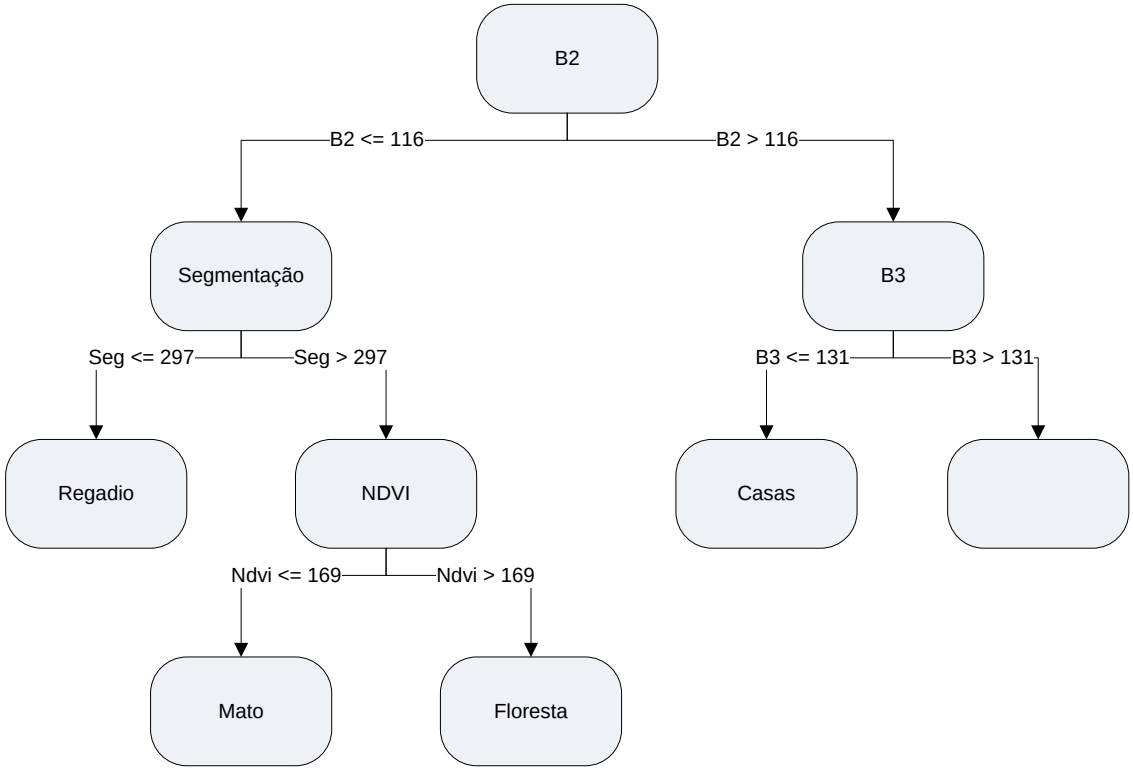
Impureza 2,55617873
racio ganho 1,594504396

Para a decisão do atributo de classe que melhor se ajusta para a divisão do nó, usou-se uma nova medida, o rácio do ganho, uma vez que o valor da impureza calculado para cada candidato é igual em vários candidatos. Alguns algoritmos utilizam esta medida, porque as medidas de impureza através dos índices de Gini e da Entropia, tendem a favorecer atributos de classe que tenham um grande número de valores distintos. Esta medida altera o critério de divisão do nó, para ter em conta o número de resultados obtidos na hipótese de dividir os dados pelo atributo em teste. Este rácio do ganho é usado por exemplo pelo algoritmo C4.5. A formula do rácio do ganho é determina por:

Rácio Ganho = Ganho / Entropia resultante da Divisão

O atributo de classe que apresenta um menor rácio é o escolhido e neste caso divide-se novamente a tabela pelo atributo B3.

A árvore de decisão é organizada da seguinte forma:



A tabela de dados é dividida com exclusão do atributo de classe B3.

Classes Solo	NDVI	B1	B4	Segmentação
estradas	10	158	139	963
estradas	13	159	144	816
estradas	17	175	154	991
soloadescoberto	70	177	168	9
soloadescoberto	72	178	170	9
soloadescoberto	76	176	171	175

Uma vez que todos os valores da subdivisão da tabela pelos vários atributos é de 1 e todos eles têm o mesmo valor de ganho informacional, então escolhe-se o primeiro atributo de classe, neste caso, o NDVI.

Classes Solo	NDVI
estradas	10
estradas	13
estradas	17
soloadescoberto	70
soloadescoberto	72
soloadescoberto	76
Entropia	1

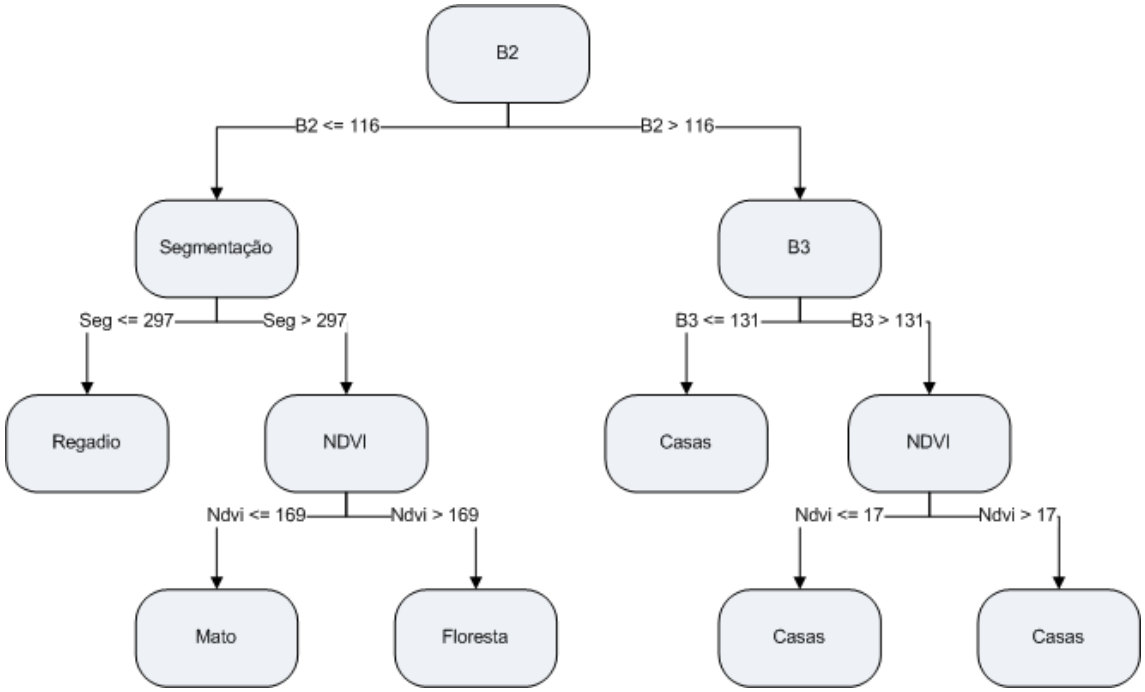
Classes Solo	B1
estradas	158
estradas	159
estradas	175
soloadescoberto	176
soloadescoberto	177
soloadescoberto	178
Entropia	1

Classes Solo	B4
estradas	139
estradas	144
estradas	154
soloadescoberto	168
soloadescoberto	170
soloadescoberto	171
Entropia	1

Classes Solo	Segmentação
soloadescoberto	9
soloadescoberto	9
soloadescoberto	175
estradas	816
estradas	963
estradas	991
Entropia	1

Chegou-se então ao fim do cálculo de todas as classes de dados atribuídas aos nós de folhas da árvore.

A árvore de decisão final é apresentada da seguinte forma:



III.4 Limitações das árvores de decisão

As árvores de decisão apresentam as limitações comuns dos algoritmos de aprendizagem, como a sobre-aprendizagem. Ou seja, as árvores de decisão são treinadas para parar quando estas classificarem perfeitamente todos os dados de treino, ou seja, cada ramo da árvore é estendido de forma a classificar correctamente todos os exemplos relevantes para aquele ramo de forma exaustiva.

Existem várias abordagens para tentar ultrapassar o problema da sobre-aprendizagem e resumem-se fundamentalmente a dois tipos: parar a árvore de decisão antes de esta chegar à perfeição ou permitir que a árvore cresça completamente e depois remover à posteriori alguns nós/ramos da árvore.

Esta segunda abordagem tem tido mais sucesso na prática. Mas ambas as possibilidades não contemplam a questão de determinar o tamanho que seria correcto para árvore de decisão.

Devido a estas limitações dos algoritmos de árvores de decisão muitos implementam uma estratégia gananciosa e assim não executam uma pesquisa exaustiva, todavia esta sequência de prioridades em geral também não é perfeita. Ou seja, as regras de decisão geradas pela árvore de decisão acabam por não ser as melhores regras.

III.5 Programas de software/Algoritmos

CART

O programa de software CART é talvez o programa mais utilizado na elaboração de árvores de decisão na área de *data mining*. É uma ferramenta que reconhece mais de 80 formatos de dados, incluindo programas estatísticos conhecidos, como o SAS ou o SPSS e permite gerar relatórios, gráficos e grafos dos resultados obtidos.

As principais características do algoritmo CART para as árvores de decisão orientadas à classificação são a criação de árvores de decisão binária, o suporte a variáveis contínuas ou pertencentes a uma categoria e a divisão dos nós durante o processo de criação em que recorre à escolha dos nós mais puros através do índice de Gini.

WEKA (C4.5)

O WEKA é um programa de software desenvolvido pela Universidade de Waikato constituído por vários algoritmos de aprendizagem, maioritariamente utilizados na área de *data mining*.

A linguagem de desenvolvimento do programa é Java e contém ferramentas nas mais diversas temáticas como pré-processamento, classificação, regressão, *clustering*, regras de associação e visualização.

Um dos algoritmos que implementa é o já conhecido C4.5 desenvolvido por Ross Quinlan. Este algoritmo é muito idêntico ao algoritmo que serviu de demonstração da geração da árvore decisão, detalhado no capítulo III.

A criação das árvores de decisão pelo programa WEKA gera um ficheiro de texto com o resultado da árvore de decisão.

RuleGen – software desenvolvido como extensão para o ENVI

O software RuleGen é uma extensão ao ENVI que implementa árvores de classificação e regressão (CART) e utiliza a ferramenta nativa do ENVI para execução de árvores de decisão.

O RuleGen utiliza algoritmos CART *freeware* que disponibilizam funcionalidades semelhantes aos algoritmos comerciais, como o CART™[2] e o See5/Cubist[3].

Os algoritmos *freeware* utilizados pelo RuleGen são o QUEST e o CRUISE. Estes algoritmos foram criados e são mantidos por dois professores/investigadores, Wei-Yin Loh da Univerdade de Wisconsin em Madison e por Yu-Shan Shih da Universidade National Chung Cheng em Taiwan. As versões utilizadas dos algoritmos com o programa RuleGen são a versão 1.9.1 do QUEST e a versão 2.2 do algoritmo CRUISE.

[2] http://www.salford-systems.com
[3] http://www.rulequest.com

QUEST

O algoritmo QUEST para a construção da árvore de decisão permite a selecção das variáveis de atributos dos dados sem viés (erro sistemático usado em estatística) ao contrário de outros algoritmos mais conhecidos e usados para a construção de árvores de decisão, como o CART.

Este algoritmo permite ainda incluir a supressão de nós durante a construção e permite a divisão binária de nós que determina que cada nó da árvore só pode ter dois nós descendentes. Para a execução do algoritmo é possível definir vários parâmetros de configuração.

O método para a divisão dos nós na construção da árvore pode ser feito através da aplicação de testes a uma única variável (univariada) ou à combinação linear de várias variáveis de atributos de classe.

Para a selecção da variável o método pode ser feito através de testes estatísticos que evitam o viés ou através do método de pesquisa exaustiva, sendo este método o mesmo usado no algoritmo CART.

O método de pesquisa exaustiva usa como critério de divisão das variáveis nos nós o índice Gini, em vez dos métodos de divisão dos nós descritos anteriormente.

O método de combinação linear das variáveis na divisão dos nós é significativamente melhor em termos de precisão e no tamanho da árvore gerado. O seu tempo de computação é muito menor quando comparado com a pesquisa exaustiva, mas mesmo assim é superior ao método de uma única variável. Este por sua vez apresenta precisão semelhante aos resultados da pesquisa exaustiva e tende a produzir árvores que são maiores. (Loh, W.-Y. and Shih, Y.-S. 1997).

CRUISE

O algoritmo CRUISE produz árvores de decisão que podem dividir cada nó em outros nós descendentes consoante o número de classes de cada variável independente. Todavia, no programa RuleGen, existe uma forma de transformar a árvore de decisão criada pelo algoritmo CRUISE numa árvore de decisão binária de modo a funcionar no programa do ENVI. Por essa razão, este algoritmo não fará parte da metodologia aplicada nesta dissertação.

ENVI – Ferramenta de construção de árvores de decisão

O programa Envi disponibiliza uma ferramenta para criar e executar árvores de decisão. O processo de criação recorre a um editor gráfico simples para criar as árvores juntamente com os seus nós e regras de decisão. Estas podem servir-se das variáveis de entrada pretendidas que se queriam adicionar, bem como um conjunto de funções a usar disponibilizadas pela ferramenta.

As funções podem ser de várias categorias, desde funções aritméticas como a adição, a subtracção, a multiplicação e a divisão das variáveis, funções trigonométricas como o seno, coseno e a tangente, funções de operadores lógicos, desde o maior, igual, menor, e/ou, negação, máximo, mínimo ou ainda funções no contexto da detecção remota, como o ndvi, as componentes principais, o declive, a orientação ou as médias e desvio padrão das bandas da imagem.

De seguida apresenta-se uma tabela com as principais características de cada algoritmo:

Tabela 3: Tabela de Comparação dos algoritmos de classificação por árvore de decisão

	GUIDE	**QUEST**	**CRUISE**	**CART**	**C4.5**
Divisão imparcial	Sim	Sim	Sim	Não	Não
Divisões por nó	2	2	≥ 2	2	2
Detecção de interacções	Sim	Não	Sim	Não	Não
Ranking	Sim	Não	Não	Sim	Não
Classes prévias	Sim	Sim	Sim	Sim	Não

Custos de erros de classificação	Sim	Sim	Sim	Sim	Não
Divisões Lineares	Sim	Sim	Sim	Sim	Não
Divisões por categorias	Subconjuntos	Subconjuntos	Subconjuntos	Subconjuntos	Atomos
Modelos de Nós	S, K, N	S	S, L	S	S
Valores em falta	Especial	Imputação	Substituto	Substituto	Pesos
Diagramas em Texto	Texto e LATEX	Texto e LATEX	Texto e LATEX	Proprietário	Texto
Bagging	Sim	Não	Não	Não	Não
Forests	Sim	Não	Não	Não	Não

Modelos de Nós: S- Simples, K – Kernel, L – Descriminante linear, N – vizinho mais próximo

III.6 Casos de Estudo aplicados à Classificação de Imagens utilizando Árvores de Decisão

Recentemente foram realizados estudos para novos classificadores baseado em árvores de decisão para imagens de satélite Landsat servindo como base de referência. Posteriormente foram utilizadas novas imagens de satélite, de apenas quatro bandas, como por exemplo SPOT-like SRC (SSRC) ou IKONOS-like SRC (ISRC), para depois comparar a exactidão da classificação e a robustez a mudanças dos vários conjuntos de

imagens de satélite com o sistema Landsat que serve de base de referência (Baraldi, et al, 2010).

Neste estudo é apresentada uma arquitectura adoptada recentemente, definida por um sistema hierárquico de compreensão da imagem de satélite em duas camadas estratificadas para imagens Landstat.

Esta arquitectura segue um modelo de arquitectura já usado na detecção remota denominado por sistema de compreensão de imagens de satélite (RS-IUSs). O objectivo da arquitectura proposta é criar uma arquitectura RS-IUS adequada e elegível para usar em sistemas de medição operacionais de imagens de satélite previstos pelos programas GMES e GEOSS (Baraldi, et al, 2010).

Existem outros sistemas que implementam a arquitectura RS-IUS, tais como: multi-agentes híbridos que combinam mecanismos de inferência dedutivos e indutivos, através de módulos de sistemas de segmentação cujo resultado são segmentos sem qualquer conhecimento semântico.

Este sistema tem como principais limitações a insuficiência artificial causada pelo problema dos sistemas de segmentação afectados pelos erros de segmentação de omissão e de comissão.

Outro sistema bem conhecido desde os anos 80, é o sistema de duas camadas baseado em segmentos. Este sistema segue um raciocínio dedutivo, ou seja, parte de uma visão mais genérica.

Estes sistemas foram desenvolvidos pela comunidade de inteligência artificial e tem algumas limitações como a falta de flexibilidade em que as regras não se adaptam à mudança dos dados e a falta de escalabilidade quando se classificam sistemas complexos. Contudo, este tipo de sistema ganhou notoriedade para imagens de muito alta resolução, sendo o *software* comercial eCognition o mais conhecido.

Para evitar o problema da insuficiência artificial este sistema propõe uma segmentação iterativa hierárquica na primeira camada. O objectivo desta solução é gerar segmentações multi-escala para verificar se alguma delas é adequada ao resultado final. Contudo, para quantificar as segmentações em termos de qualidade seria necessário ter dados do terreno a todas as escalas geradas, o que é praticamente impossível de obter.

Desta forma, o segmento final escolhido é determinado por um conjunto de critérios heurísticos, subjectivos ou qualitativos.

O processo de segmentação hierárquica não é fácil de usar e requer uma grande ocupação de memória e de processamento.

Outro sistema, mais recente que os anteriormente descritos, denomina-se por implementação Shackelford e Davis (Shackelford, et al, 2003).

A principal característica desde sistema consiste na classificação supervisionada do algoritmo máxima verosimilhança na primeira camada, baseado no pixel, de modo a obter classes semânticas da imagem.

Cada classe obtida da classificação é uma combinação das classes reais do uso do solo, provavelmente sobrepostas espectralmente devido à confusão do algoritmo de máxima verosimilhança. Depois de obtidas as classes e identificadas como mutuamente exclusivas, estas podem ser adoptadas em mecanismo de aprendizagem como árvores de decisão de forma hierárquica. Na figura seguinte apresenta-se a arquitectura proposta por este sistema.

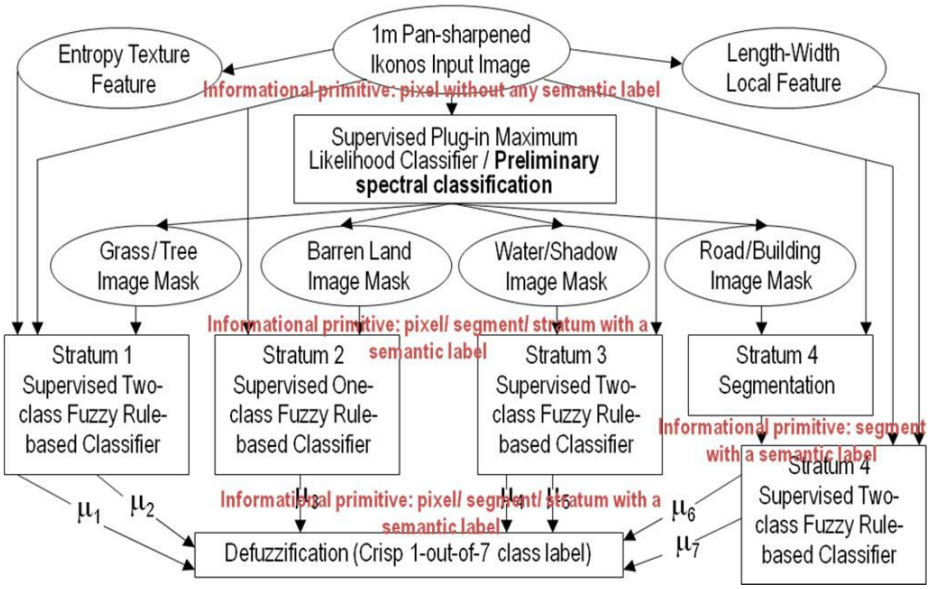


Figura 5: Sistema estratificado hierárquico de duas camadas

(Shackelford, et al, 2003)

Na segunda camada do sistema é possível estratificar a informação proveniente da classificação preliminar executada na camada anterior.

O conceito de estratificação é muito usado em estatística. A ideia subjacente consiste em dividir os dados em grupos não sobrepostos, por exemplo áreas geográficas, para garantir uma maior precisão.

Pode haver estratos de classes associados as características das classes, como textura, geometria, forma ou morfologia. Estes tipos de classes estão associadas aos objectos feitos pelo homem como edifícios e estradas do qual as suas propriedades geométricas são importantes para o seu reconhecimento.

A grande limitação desde sistema consiste na necessidade de classificação supervisionada com os dados de treino nas camadas hierárquicas (Shackelford, et al, 2003). O sistema proposto neste artigo, baseado no sistema de Shackelford e Davis, sugere a omissão da classificação preliminar com recurso ao algoritmo de máxima verosimilhança na primeira camada.

O que distingue este sistema dos sistemas anteriores é que no primeiro nível/camada os objectos estão identificados semanticamente. Este tipo de abordagem oferece a capacidade de detectar pequenos detalhes mas genuínos da imagem que são potencialmente superiores aos segmentos gerados pelo algoritmo de segmentação. Por esta razão, ao trabalhar directamente na resolução, esta abordagem de mapeamento é independente de qualquer sensor de imagens de satélite (Baraldi, et al, 2010).

Uma das análises prende-se em comparar e avaliar, neste caso, a dicotomia entre vegetação e não vegetação para o sistema de Landsat e o sistema Spot. Ou seja, avaliar os resultados quando se aumenta em termos espaciais mas baixa-se em termos espectrais, como é o caso do sistema Spot.

O resultado foi muito satisfatório e muito equivalente ao resultado produzido pelo sistema Landsat 7 ETM+, todavia com uma perda de informação de cerca 1,5%.

O estudo comprova que os sistemas de classificação propostos são efectivos, executam quase em tempo real, são completamente automatizados e robustos a mudanças nos dados adquiridos ao longo do tempo, do espaço e dos sensores (Baraldi, et al, 2010).

A implementação do sistema através de duas camadas estratificadas, em que a primeira camada é uma classificação preliminar baseada no pixel, dá origem a resultados finais superiores na exactidão da classificação quando comparado com uma

implementação não estratificada, como por exemplo os sistemas que são constituídos apenas por uma camada.

Outro estudo realizado por (Abdelhamid et. al., 2010) teve como objectivo criar mapas de salinidade dos solos a partir de árvores de decisão. Utilizaram-se dois índices de vegetação, o NDVI e o SAVI, para detectar a presença de salinidade uma vez que estes índices são óptimos para a detecção de vegetação e uma vez não encontrada, por exclusão de partes, significaria que estaríamos perante áreas de salinidade. Foi também usado o índice de TCT constituído por valores de brilho, vegetação e humidade que serviram para distinguir áreas com grandes valores de refletância espectral, áreas verdes e humus no solo.

Os rácios das bandas B3/B1 e B5/B7 foram calculados para interpretar algumas propriedades associadas a solo não salinizado, pois o primeiro rácio ajuda a determinar áreas onde o material aço é reflectido. Já o segundo rácio tem uma forte correlação com materiais minerais em áreas com pouca vegetação.

Por fim utilizaram-se também, para as áreas em análise, valores de declive e orientação. Seguindo estes pressupostos, procedeu-se à criação da árvore de decisão utilizando o algoritmo C4.5.

A classificação por árvore de decisão provou ser eficiente e útil para áreas bastante grandes quando comparadas com outras técnicas de detecção remota. A árvore de decisão incorporou diversas variáveis de ambiente que influenciaram significativamente os mapas de salinidade produzidos.

Outro estudo realizado por Bouziani, et. al., 2010, teve como objectivo desenvolver um algoritmo de segmentação e um classificador baseado em objectos de forma a ajudar no processo de classificação de imagens multi-espectrais de muito alta resolução em áreas urbanas.

Por sua vez, as áreas em estudo recorrem a imagens de satélite multi-espectrais IKONOS. Neste caso a metodologia envolveu primeiro uma fusão de imagens pancromáticas com multi-espectrais para obter uma resolução de 0.6 metros no pixel. A partir desta imagem gerada, procedeu-se à classificação baseada no pixel com o

algoritmo máxima verosimilhança e à segmentação para obter grupos homogéneos de pixéis.

Estes dois resultados e complementado com uma base de regras serviram de variáveis de entrada para a construção de um classificador baseado em regras de objectos.

Para a validação de resultados criaram-se áreas de teste e de treino que não se intersectassem e usaram-se algumas variáveis de informação para reduzir a confusão espectral entre as classes.

A textura foi uma das variáveis que ajudou a diferenciar entre árvores e erva. Normalmente estas classes têm assinaturas espectrais semelhantes mas texturas diferentes, concretamente, a textura da classe de erva é mais homogénea.

Outras variáveis de informação que se utilizaram foram a natureza geométrica da classe e a informação de contexto, permitindo fazer a distinção entre edifícios e estradas. Também identificaram o facto de alguns objectos gerarem sombra e outros não, o que pode ajudar a separar classes que sejam espectralmente idênticas.

De modo a utilizar estes atributos, foi realizada uma abordagem orientada ao objecto, nomeadamente a segmentação de imagem multi-espectral (Bouziani, et. al., 2010).

Para a criação do algoritmo de segmentação utilizou-se o algoritmo de crescimentos de regiões na imagem com definição de limites espectrais de forma a obter objectos homogéneos, bem como informação geométrica e análise de vizinhança dos segmentos. Após a imagem estar segmentada calcularam-se atributos geométricos e espectrais para cada segmento.

Para o classificador a gerar utilizou-se a classificação obtida pelo algoritmo máxima verosimilhança e analisaram-se as classes que pertenciam aos segmentos. Recorreram-se a vários atributos geométricos para identificar os segmentos às classes respectivas e vice-versa, para a determinação de regras a usar para o classificador. Os resultados obtidos pelos métodos propostos, foram consideravelmente melhores quando comparados com a classificação de máxima verosimilhança. Futuras melhorias passam por criar um método de segmentação temático, em que não seja necessário criar áreas de treino.

Capitulo IV: Metodologia

A metodologia utilizada na presente dissertação consiste em descrever as várias etapas, desde a realização das áreas de treino e de teste até à classificação de imagens com recurso aos algoritmos de árvores de decisão.

De seguida são descritos todos os passos envolvidos na metodologia proposta, bem como os resultados dos vários ensaios para a extracção de classes de ocupação de solo a partir de dados digitais aéreos de alta resolução.

IV.1 Segmentação

Para a segmentação da imagem foi usado o software *open source* SPRING.

Por sua vez, o método utilizado no processo de segmentação foi o Crescimento por Regiões.

Neste método existem duas medidas a definir, a Similaridade e a Área em pixels.

A Similaridade é baseada na distância Euclidiana entre os valores médios dos níveis de cinza de cada região. Ou seja, duas regiões são consideradas distintas se a distância entre as suas médias for superior ao limite da medida de Similaridade.

Por outro lado, as regiões cuja área é menor que o mínimo escolhido são agrupadas pelas regiões adjacentes mais similares a estas.

Foram realizados vários testes de valores de Similaridade e de Área para determinar a melhor imagem cujos segmentos gerados mais se aproximam dos objectos reais da imagem.

O objectivo é produzir objectos similares, logo o limite definido para o parâmetro de Similaridade não deverá ser muito elevado. Dada a grande resolução da imagem, pretende-se que os segmentos tenham uma área relativamente grande, ou seja, o suficiente para tentar segmentar de uma forma homogénea os objectos reais da imagem. De seguida apresentam-se alguns exemplos dos vários níveis obtidos dos segmentos da imagem, consoante os parâmetros definidos, para demonstrar as diferenças de cada resultado obtido e perceber a razão dos valores escolhidos para os parâmetros do algoritmo de segmentação de imagem.

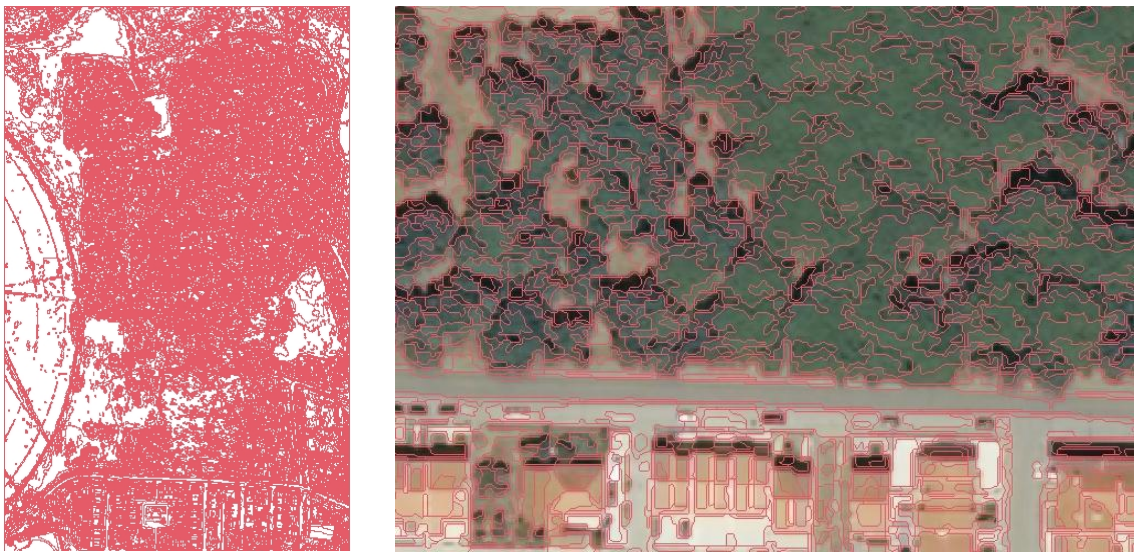


Figura 6: Exemplo de segmentação - Similaridade 10, Área 10 pixels

O parâmetro de Similaridade seleccionado foi de 10 e o de Área foi nomeadamente de 10 pixels. Como se pode ver pelas figuras anteriores foram gerados muitos segmentos e a imagem como um todo está muito segmentada face aos objectos reais que se pretendem classificar na imagem.

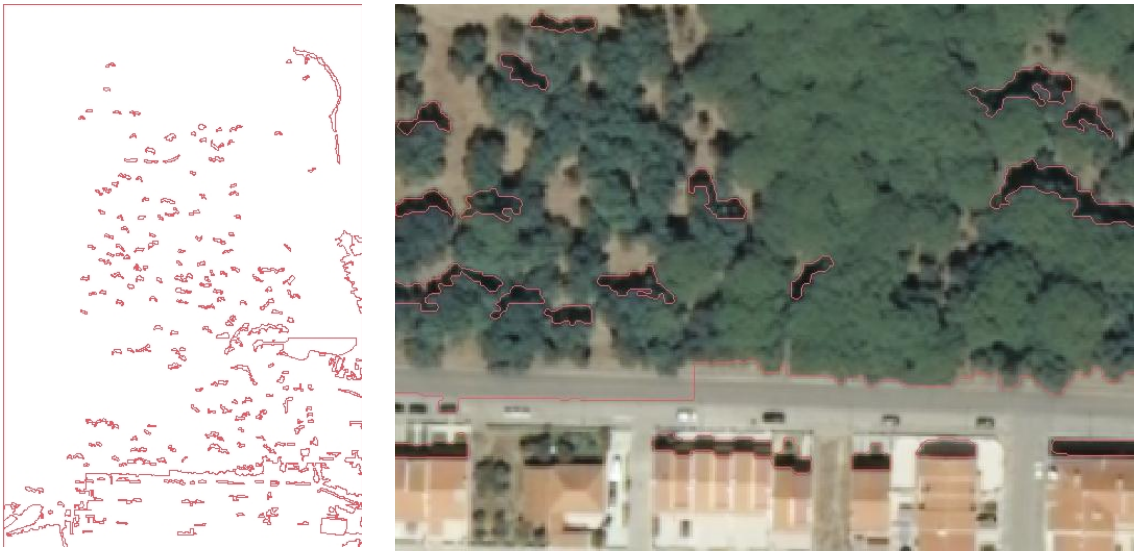


Figura 7: Exemplo de segmentação - Similaridade 100, Área 100 pixels

O parâmetro de Similaridade seleccionado foi de 100 e o de Área foi nomeadamente de 100 pixels. Neste caso, as figuras anteriores demonstram que foram gerados poucos segmentos, focando-se principalmente naqueles cujos pixels de cinza eram mais escuros, uma vez que a distância entre a média das regiões dos pixels era muito superior. Este resultado representa pouco os objectos reais que se pretendem classificar na imagem.

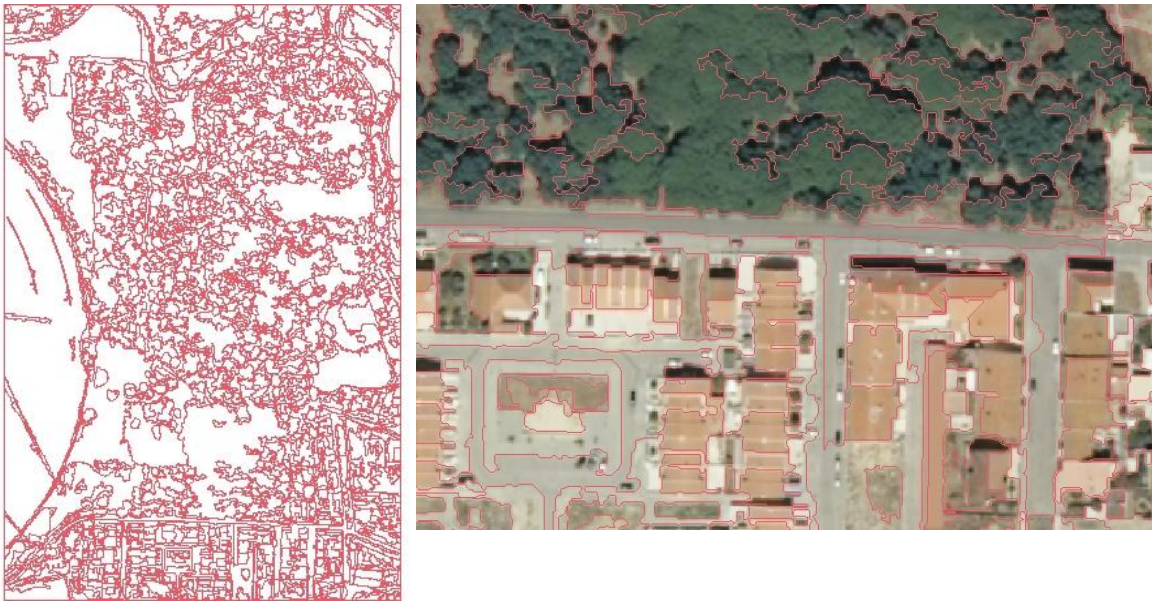


Figura 8: Exemplo de segmentação - Similaridade 15, Área 300 pixels

O parâmetro de Similaridade seleccionado foi de 15 e o de Área foi nomeadamente de 300 pixels. A figura mais à esquerda revela visualmente as áreas e os segmentos que mais se assemelham aos objectos reais da imagem. Por essa razão foram escolhidos estes parâmetros, contudo, foram realizados na mesma testes de classificação com as segmentações anteriores para comprovar e a avaliar a escolha destes parâmetros.

IV.2 Regiões de Interesse de referência

As Regiões de Interesse de referência foram aplicadas para um segmento de imagem previamente criado a partir de uma fotografia aérea digital de muito alta resolução na região do Montijo.

As classes identificadas para as Regiões de Interesse partiram de uma análise prévia do Corine Land Cover 2006 e suas classes de uso de solo. Consequentemente, foram contempladas para o segmento em estudo quatro classes base: Culturas Temporárias de Regadio, Salinas e aquicultura litoral, Florestas de folhosas e Sistemas culturais e parcelares complexos, identificadas na figura seguinte:

Figura 9: Classes Corine Land Cover 2006 para o segmento de imagem em estudo

As classes identificadas pelo Corine serão utilizadas como classes das áreas de referência, todavia, e devido ao nível de detalhe pretendido, são identificadas mais duas classes, designadamente: estradas e casas, que o Corine descreve como Industria, comercio e equipamentos gerais e Tecido urbano contínuo e Tecido urbano descontínuo.

As Regiões de Interesse de referência foram criadas a partir da foto interpretação da imagem com recurso ao software de edição/criação de vectores *Arc Map*.

A figura seguinte apresenta o resultado final, cujo objectivo foi identificar as classes de uso de solo mais importantes e distinguir as classes de uso do solo como espectralmente idênticas:

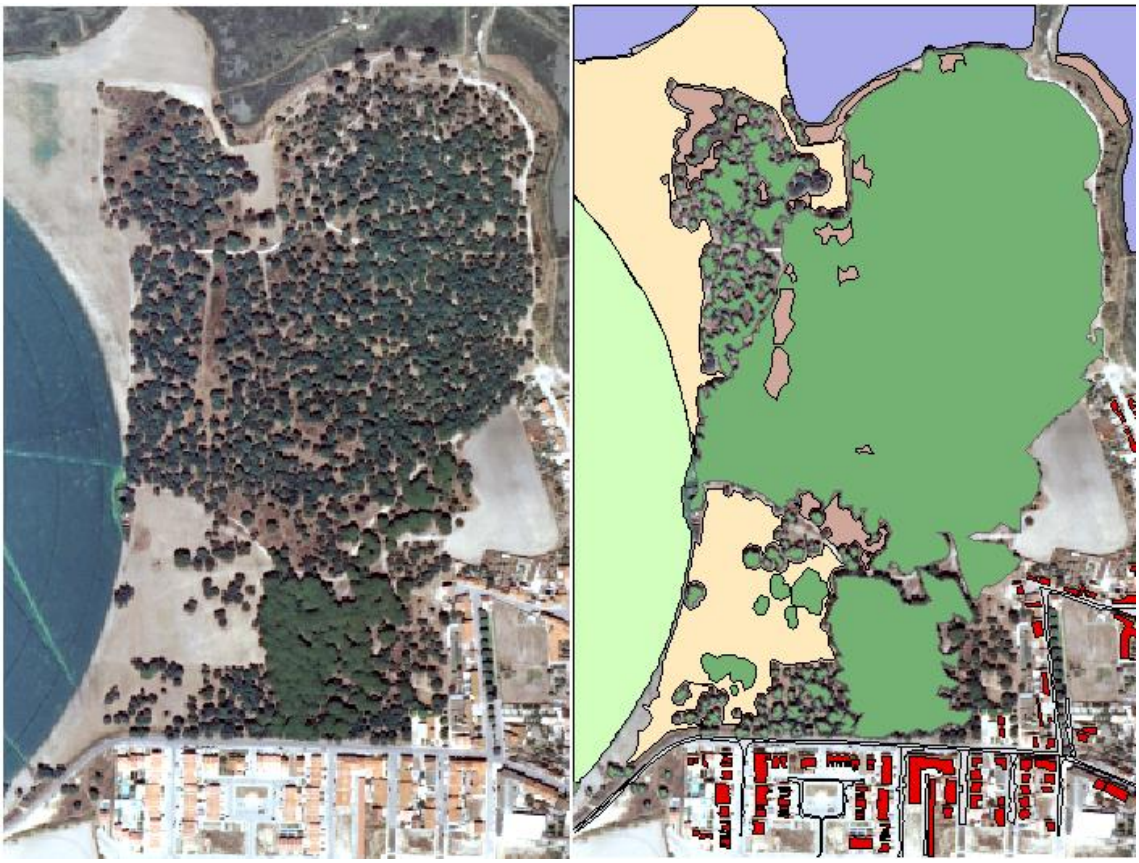


Figura 10: Regiões de Interesse de referência e Imagem de Referencia

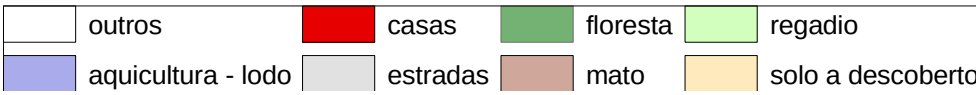


Figura 11: Legenda das Regiões de Interesse de Referencia

Todos os polígonos criados para as regiões de interesse foram unificados por tipo de categoria para que cada região de interesse seja univocamente identificada. Foi também criado um novo campo “tipo”, para identificar o tipo de região de interesse.

Tabela de Atributos

FID	Id	tipo
0	0	regadio
2	0	aquicultura - lodo
121	0	casas
224	0	estradas
244	0	mato
1	0	solo a descoberto
243	0	floresta

Figura 12: Tabela de Atributos das Regiões de Interesse de referência

A partir do ficheiro *shapefile* gerado para a construção das Regiões de Interesse criaram-se as regiões de interesse no formato de ficheiro *evf* (formato este usado no programa ENVI para determinar as regiões de interesse).

IV.3 Técnicas com Árvores de Decisão

Os parâmetros de configuração do algoritmo QUEST que apresentaram valores por defeito para a realização dos testes foram:

- O valor mínimo de registos na amostra de dados para a divisão dos nós durante a construção da árvore: 5 (quanto menor é este valor, maior fica a árvore gerada – o valor por defeito é obtido pelo máximo entre (5 e n/100) onde n é número de total de registos da amostra de dados);

- O valor que permite controlar o tamanho da árvore de decisão, de forma a evitar a sobre-aprendizagem (*overfitting*) : 0. O valor de zero, retorna a árvore com um valor mínimo estimado para a validação de erros ou custos da classificação.

- O valor que define o número de vezes que é executada a validação: 10. Este valor é o recomendado e usado pela maior parte dos algoritmos. Se o valor for superior o tempo de cálculo da árvore de decisão aumenta. Esta medida é denominada na literatura pelo nome *v-fold-cross-validation* como uma técnica de validação que consiste em gerar árvores de decisão com dados aleatórios para cada tamanho da árvore gerada. Estas árvores de decisão são depois validadas com amostras de teste com o objectivo de obter a melhor precisão média das classificações previstas.

IV.4 Arquitectura do Problema

A arquitectura do problema proposta e descrita na presente dissertação centra-se fundamentalmente na criação da árvore de decisão que melhor satisfaça os resultados pretendidos quanto a exactidão e precisão da classificação de classes de uso de solo em imagens de muito alta resolução.

De seguida apresenta-se a arquitectura proposta para o problema e descrevem-se os passos, de forma sequencial e genérica, para alcançar os objectivos propostos.

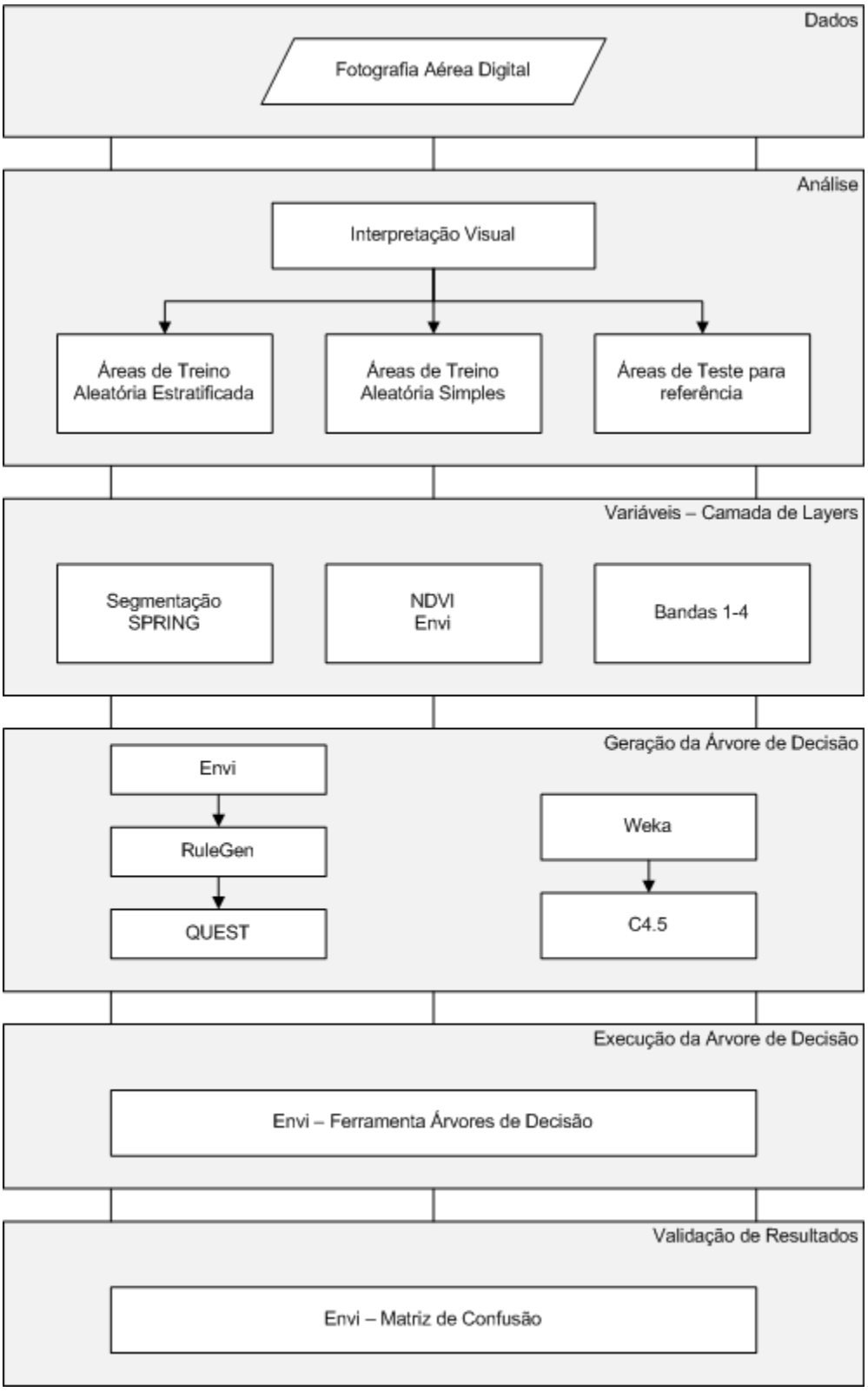


Figura 13: Arquitectura da solução proposta

Passo1: Geração das Áreas de Treino de forma aleatória e simples

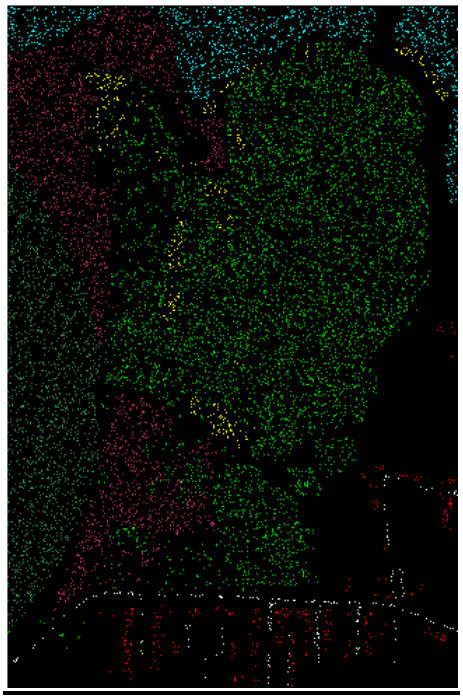


Figura 14: Área de Treino aleatória

Para gerar as áreas de treino, recorreu-se à geração automática e aleatória disponibilizada pelo programa ENVI, a partir das regiões de interesse de referência. Foi usado o método Geração Aleatória Estratificada (*Random Stratified*) proporcional, em que se atribui a cada classe de teste uma percentagem de 10% a extrair de cada classe das regiões de interesse de referência.

Legenda de classes/cores						
regadio	lodo	solo	mato	floresta	casas	estradas

Figura 15: Legenda de cores das classes

Figura 16: Área de treino simples

Esta é uma amostra simples com vários polígonos gerados manualmente para cada classe, onde se procurou identificar univocamente cada classe sem haver sobreposição entre as classes.

O objectivo passa por tentar comparar os resultados entre as amostras para cada classificação de modo a verificar se estes conseguem separar as classes que se distinguem visivelmente.

Passo2: Geração do NDVI

A geração do NDVI, índice de vegetação normalizado, constitui um índice de referência muito usado para a classificação de imagens em Detecção Remota, principalmente quando as imagens a classificar são constituídas por classes relacionados com vegetação. Por essa razão, os valores do índice de vegetação normalizado vão ser usados como variáveis independentes na construção da árvore de decisão.

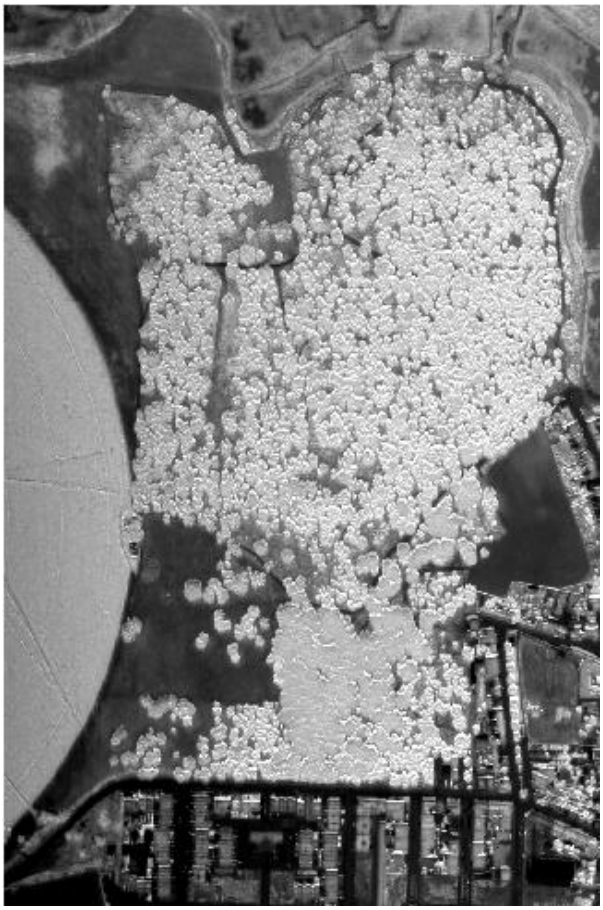


Figura 17: NDVI gerado a partir da imagem

Passo3: Criação da camada de variáveis para a geração da árvore de decisão

A camada de variáveis para a geração da árvore de decisão partiu da funcionalidade do programa ENVI, de criar uma camada de *layers* (*layer stacking*) para servir de entrada para a criação da árvore de decisão. As variáveis usadas foram: todas as bandas da imagem de referência de muito alta resolução, isto é, as bandas do espectro visível e a banda do infravermelho próximo; a imagem do índice de vegetação normalizado e a imagem proveniente da segmentação obtida através do programa SPRING.

Depois de criada a camada de layers são criadas ou importadas as áreas de treino, que vão ser a amostra dos dados para a criação da árvore de decisão.

Passo4: Geração da árvore de decisão

A geração da árvore de decisão foi realizada a partir de uma extensão desenvolvida para o programa ENVI denominada *RuleGen.*

Esta extensão permite a utilização de dois tipos de algoritmos para a criação da árvore de decisão, entre o QUEST e o CRUISE, já descritos nos capítulos anteriores.

Para cada um dos ensaios realizados, resultou um ficheiro que representa a árvore de decisão na sintaxe de linguagem que o ENVI interpreta.

O principal algoritmo utilizado foi o QUEST e foram realizados testes com algumas variações nos parâmetros de configuração para ser possível identificar a origem das diferenças e o que interfere em termos de resultados práticos.

Neste caso concreto, o algoritmo QUEST demorou em média cerca de uma hora para gerar a árvore de decisão. Quando utilizado o método linear de combinação de variáveis para a divisão dos nós o processo demora o dobro do tempo, mas o tamanho da árvore de decisão também foi menor.

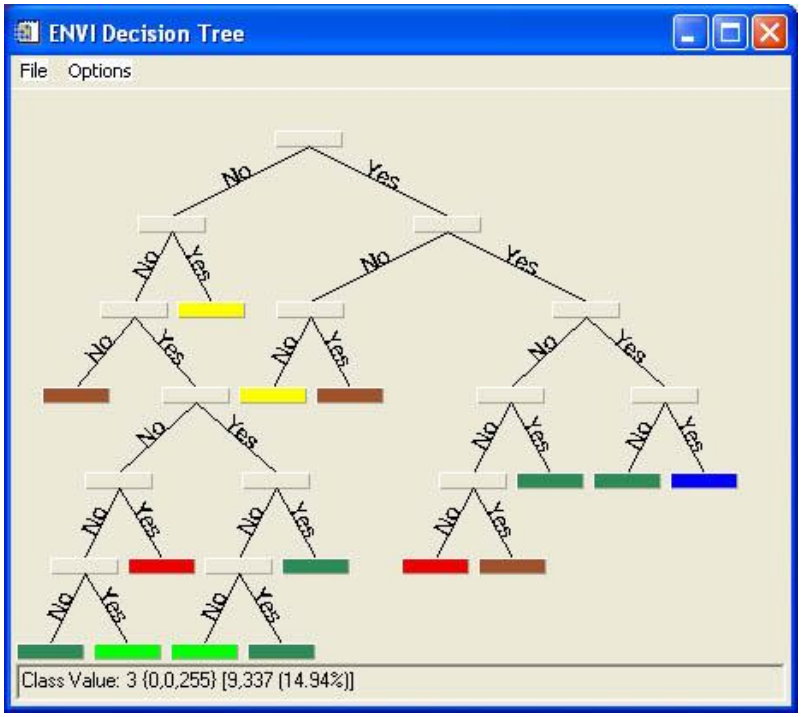


Figura 18: Exemplo de desenho da uma árvore de decisão no ENVI gerado pelo QUEST.

De forma a calcular a árvore de decisão através do algoritmo C4.5 utilizou-se o programa WEKA.

Este programa implementa o algoritmo C4.5 na linguagem de programação Java e aceita ficheiros de dados com uma sintaxe definida para execução do algoritmo.

Neste caso foi implementada uma interface de programação que converte os ficheiros da amostra de dados gerados pelo Programa RuleGen num ficheiro de dados cuja sintaxe é interpretada pelo o algoritmo C4.5 do Weka.

A figura seguinte demonstra o processo envolvido para geração do algoritmo C4.5.

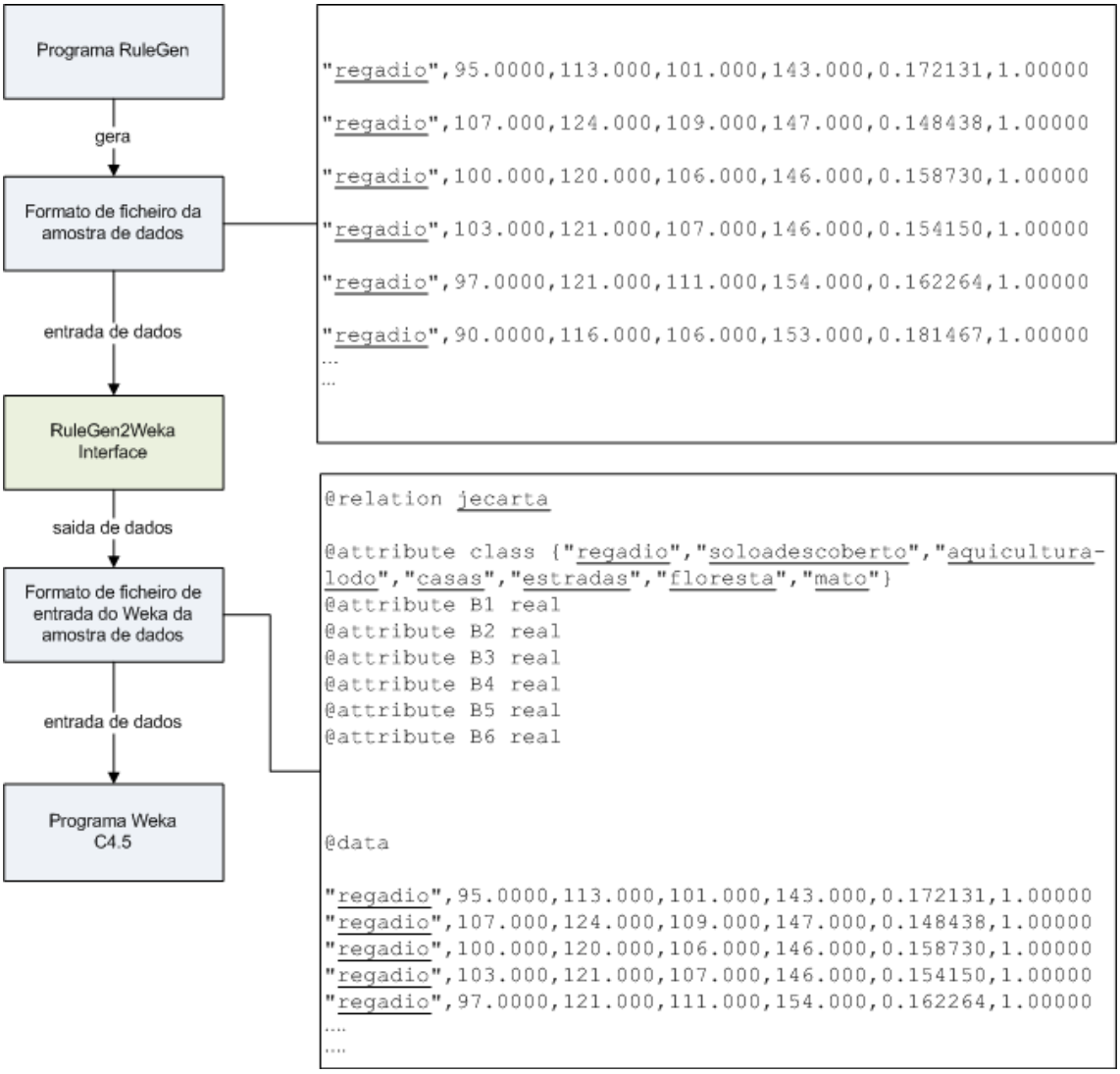


Figura 19: RuleGen2 Interface de Programação desenvolvida

Após a geração do algoritmo de árvores de decisão C4.5 é criado um ficheiro resultado com uma sintaxe definida.

De modo a utilizar a árvore de decisão para a classificação de imagens, foi desenvolvida uma nova interface que converte o ficheiro da árvore de decisão gerada pelo programa Weka num ficheiro de árvores de decisão que é interpretado pelo programa Envi. Ou

seja, o formato gerado cumpre a sintaxe estabelecida no programa Envi para representação de árvores de decisão.

A figura seguinte representa o mecanismo desenvolvido.

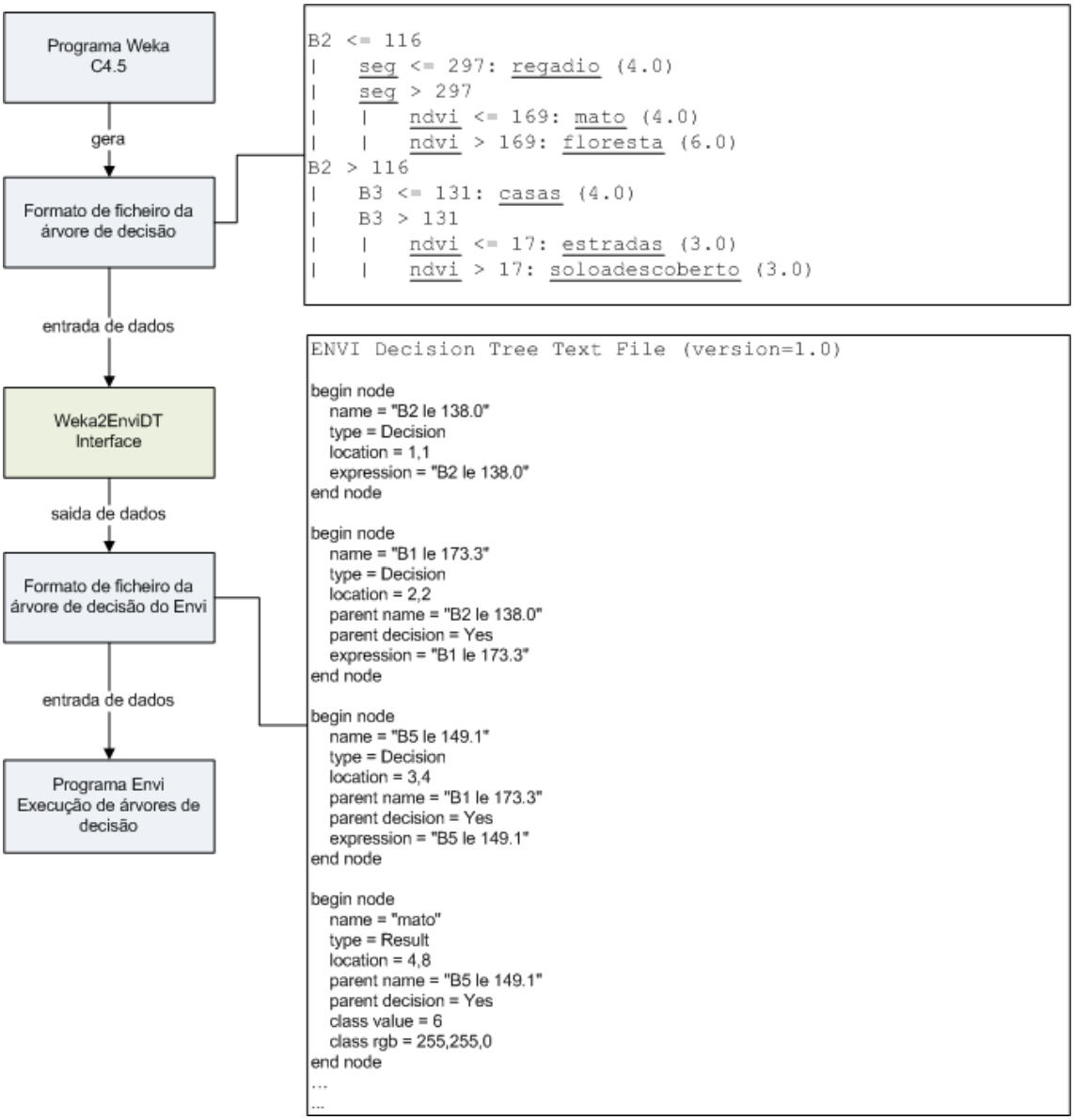


Figura 20: Weka2EnviDT interface de programação desenvolvida

Passo5: Execução da árvore de decisão

A execução da árvore de decisão é realizada a partir do programa ENVI. Os parâmetros de entrada são o ficheiro da árvore de decisão obtido no passo anterior e as variáveis de entrada (camada de *layers*) são as usadas para a construção da árvore.

O programa interpreta o ficheiro da árvore de decisão, construído de acordo com a linguagem de implementação e processa todos os nós da árvore com as respectivas regras de decisão de forma a obter uma classificação da imagem.

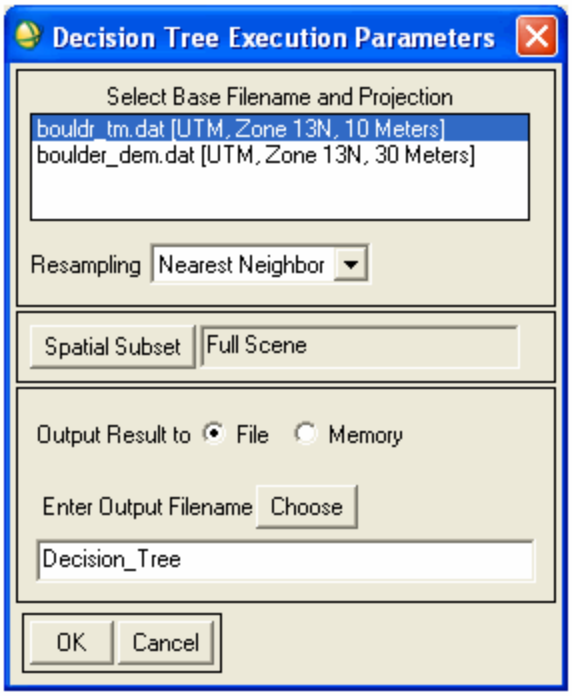


Figura 21: Execução da árvore de decisão

Passo6: Validação dos Resultados

Para validar os resultados recorre-se à Matriz de Confusão também disponibilizada pelo programa ENVI e são usadas as Regiões de Interesse de referência para validar os dados classificados.

Para além da matriz são também obtidos os seguintes dados: precisão global em termos de percentagem, o coeficiente KHAT (kappa), os erros de comissão complementares à precisão do utilizador e os erros de omissão, complementar à precisão do produtor.

A bibliografia, sugere que não haja apenas uma medida de exactidão para aceitar ou rejeitas a classificação, mas sim um conjunto de vários índices de qualidade que devem ser postos em prática e calculados (Congalton, et al, 1999 e Foody, et al, 2002).

IV.5 Resultados

Os resultados apresentados demonstram as várias experiências que foram realizadas, no decurso da presente dissertação e que visam estabelecer comparações entre os vários ensaios de forma a interpretar os resultados obtidos e reconhecer quais os que melhor resultados obtêm.

Para validação dos resultados e avaliação da qualidade da informação extraída, segue-se uma metodologia bem conhecida na área da classificação de imagens em detecção remota que concilia a matriz de confusão e o índice Kappa. São também apresentados outros factores que determinam a exactidão da informação extraída, bem como o erro associado à mesma.

Através da precisão do produtor é permitido saber quantos elementos identificados no terreno de uma determinada classe de uso de solo são também identificados no resultado produzido. Por sua vez, a precisão do utilizador, permite avaliar de entre os elementos classificados no mapa, quais os que foram identificados correctamente de acordo com os dados do terreno.

Em suma, a precisão do produtor é complementar do erro de omissão e a precisão do utilizador é complementar do erro de comissão.

1. **Execução:**

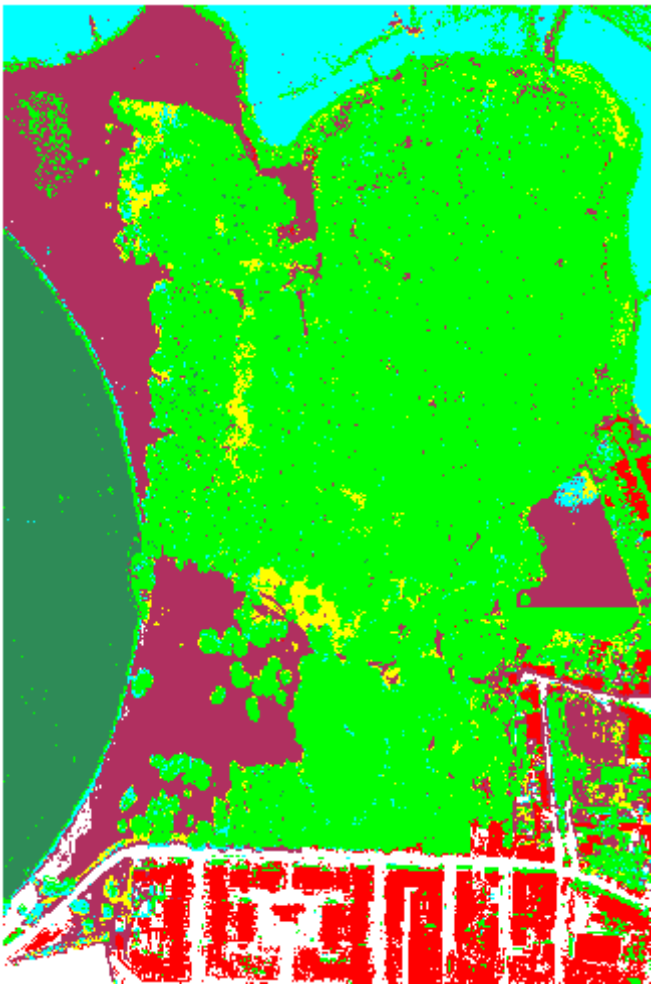
   Este ensaio serve apenas de referência, para comprovar os valores de concordância de todas as experiências.

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	100	100	MAX Veros.	Não Obtido

O resultado obtido foi apenas uma classe preenchida para toda a imagem classificada.

2. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	100	100	QUEST univariado	93,2241%



P.G.: 93.2241% Kappa: 0.8984	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	180095	0	0	1400	0	151	8	181654	99.14	0.86
casas	0	32476	48	104	9	140	0	32777	99.08	0.92
estradas	0	130	19977	17	0	370	0	20494	97.48	2.52
floresta	3922	188	109	619088	20427	10213	13354	667301	92.77	7.23
mato	0	41	20	6617	15846	924	520	23968	66.11	33.89
solo a desc.	157	376	1017	11349	2125	212799	1514	229337	92.79	7.21
aq.lodo	149	0	104	7147	562	1536	96034	105532	91.00	9.00
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	97.71	97.79	93.90	95.88	40.66	94.10	86.18			
E.O. (%)	2.29	2.21	6.10	4.12	59.34	5.90	13.82			

O seguinte ensaio consistiu na utilização de áreas de treino geradas aleatoriamente, no valor de 10% face às áreas de referência. Sendo a variável segmentação um dos atributos identificados que influencia a classificação, foram usadas nesta experiência os valores de 100 para a Similaridade e 100 pixels para a área dos segmentos. Estes valores podem, por exemplo, identificar os segmentos/objectos na imagem que sejam mais similares em distâncias maiores sempre relativamente aos níveis de cinza de cada região.

Na tabela seguinte verifica-se que a precisão global foi de 93,2241% e o índice Kappa de 0,8984. Estes valores são bastante elevados relativamente à verdade real da exactidão do classificador, pois a amostra de treino partiu de geração aleatória dos dados de referência.

Nos ensaios seguintes são utilizadas outras áreas de treino que partiram de uma amostra simples de modo a avaliar a exactidão dos classificadores e compará-los entre si. Neste caso procedeu-se apenas à análise das classes identificadas.

Os resultados obtidos para os vários atributos foram bastante aceitáveis na sua globalidade, todavia, apenas a classe de mato, teve erros de omissão acima dos 50%. Este facto deve-se essencialmente á posição da classe mato que está praticamente incluída na classe de floresta.

Uma vez que os resultados da segmentação, detalharam muito poucos segmentos similares, pois a distância especificada entre eles era bastante superior, pode ter levado a que a classe mato fosse pouco identificada em termos de segmentos de objectos.

A classe aquicultura – lodo é a classe que a seguir à classe mato apresenta maior erro de omissão.

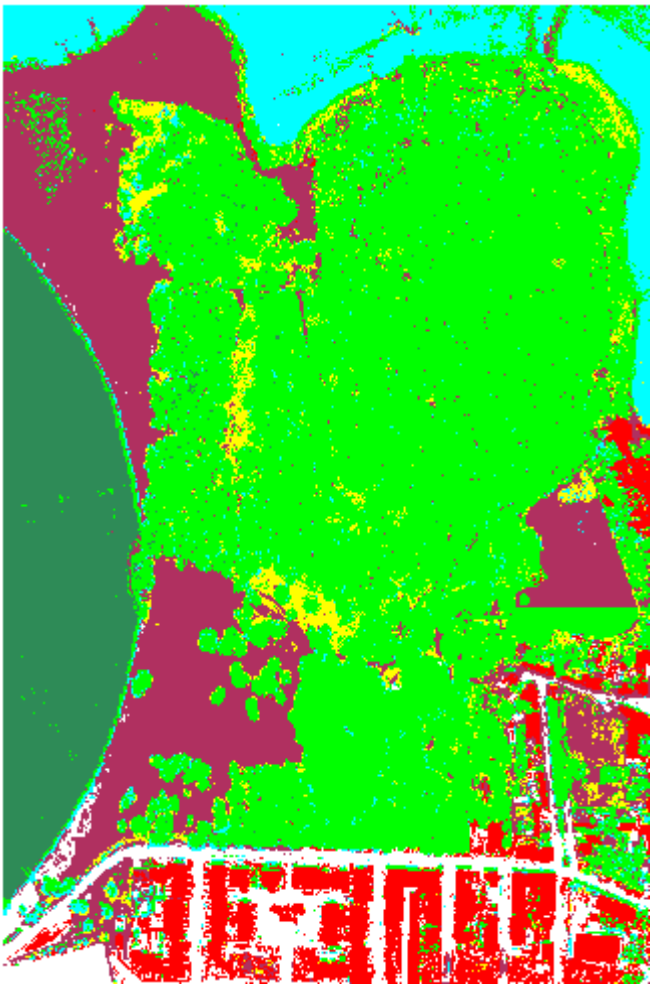
Os erros encontram-se distribuídos por outras classes, sendo os mais significativos os associados às classes de floresta, o mato e o solo a descoberto.

Este situação também pode estar relacionada não só com o facto de não haver segmentos que identifiquem na realidade a classe lodo, mas também por ser uma classe identificada nas áreas de referência como sendo uma área de grande dimensão e por essa razão agrupar níveis espectrais idênticos às identificadas.

O algoritmo QUEST mostrou ser eficiente na geração da árvore de decisão e o tempo de cálculo da árvore foi de aproximadamente uma hora. A execução da árvore de decisão para a posterior geração da classificação da imagem demorou menos de cinco minutos.

3. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	100	100	C4.5	93.5336%



P.G.: 93.5336% Kappa: 0.9037	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	180375	0	0	1463	0	164	10	182012	99.10	0.90
casas	0	32566	58	104	33	154	0	32915	98.94	1.06
estradas	0	83	20242	22	2	387	19	20755	97.53	2.47
floresta	3651	138	32	613522	16133	8145	10964	652585	94.01	5.99
mato	0	70	3	10811	20253	1529	485	33151	61.09	38.91
solo a desc.	218	353	808	11578	2028	214216	1608	230809	92.81	7.19
aq.lodo	79	1	132	8222	520	1538	98344	108836	90.36	9.64
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	97.86	98.06	95.14	95.01	51.97	94.73	88.26			
E.O. (%)	2.14	1.94	4.86	4.99	48.03	5.27	11.74			

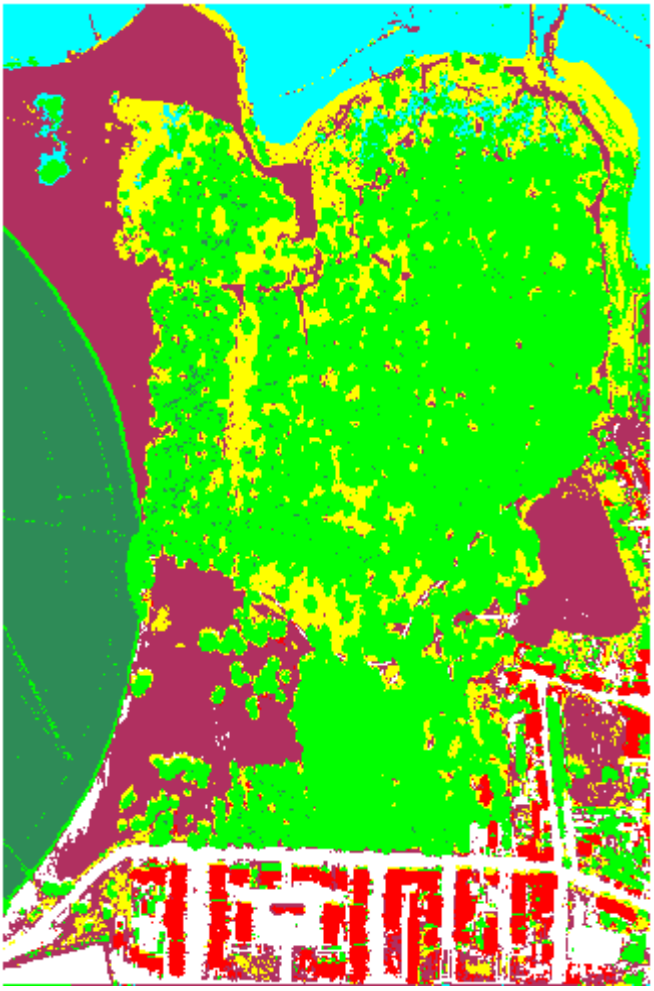
A classificação com recurso ao algoritmo de geração automática de árvore de decisão C4.5 mostrou ser mais preciso comparativamente com o algoritmo QUEST.

A precisão global aumentou, principalmente pelo aumento das classes, em geral, relativamente à precisão do produtor. Ou seja, as áreas identificadas como referência foram classificadas no resultado em maior número, comparativamente com a classificação anterior. Sendo que o número de nós do algoritmo C4.5 é, em média, o dobro do algoritmo QUEST, esperavam-se níveis de resultados melhores. Contudo não foram aplicadas validações *à posteriori* o que poderia comprometer o resultado.

Uma vez que o tamanho da árvore não é um factor relevante para o estudo em causa, este pode influenciar para classificações de imagens com uma grande quantidade de dados.

4. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	10	10	MAX Veros.	89,3873%



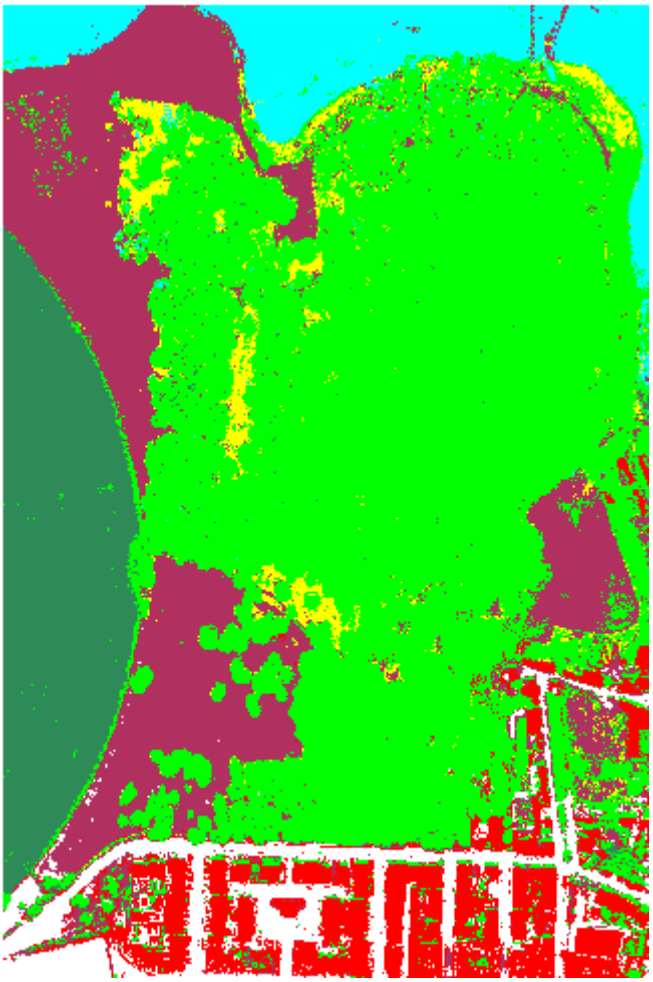
P.G:89,3873% **Kappa:**0,85	**Terreno**									
Classificação	**regadio**	**solo a desc.**	**aq.lodo**	**casas**	**estradas**	**floresta**	**mato**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	178676	14	0	0	0	5610	0	184300	**96.95**	**3.06**
solo a desc.	1	210929	1504	206	195	17543	2311	232689	**90.65**	**6.72**
aq.lodo	0	2296	106443	0	0	7529	362	116630	**91.27**	**4.48**
casas	0	287	0	32653	81	144	1	33166	**98.45**	**1.68**
estradas	0	1396	0	55	20880	637	1	22969	**90.91**	**1.86**
floresta	5646	5214	1322	75	24	542967	1612	556860	**97.51**	**15.91**
mato	0	5997	2161	222	95	71292	34682	114449	**30.30**	**11.00**
Total	184323	226133	111430	33211	21275	645722	38969	**1261063**		
P.P. (%)	**96.94**	**93.28**	**95.52**	**98.32**	**98.14**	**84.09**	**89.00**			
E.O. (%)	**3.05**	**9.35**	**8.73**	**1.55**	**9.09**	**2.49**	**69.70**			

A classificação a partir do método da máxima verosimilhança é calculada para servir como base de referência em termos de resultados para os outros ensaios que se seguem, nomeadamente as árvores de decisão.

Contudo, existem já diversos estudos que procuram comparar este método com outros que surgem cada vez mais na área da detecção remota.

5. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	10	10	QUEST univariado	96.5861%



P.G.: 96,5861% Kappa: 0.9492	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	181102	1	0	1364	0	126	1	182594	99.18	0.82
casas	0	32724	83	80	10	141	0	33038	99.05	0.95
estradas	0	63	20694	7	0	187	0	20951	98.77	1.23
floresta	3051	237	93	629292	11036	5058	1522	650289	96.77	3.23
mato	0	8	1	6720	26051	1131	360	34271	76.01	23.99
solo a desc.	170	178	404	6865	1319	219011	410	228357	95.91	4.09
aq.lodo	0	0	0	1394	553	479	109137	111563	97.83	2.17
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	98.25	98.53	97.27	97.46	66.85	96.85	97.94			
E.O. (%)	1.75	1.47	2.73	2.54	33.15	3.15	2.06			

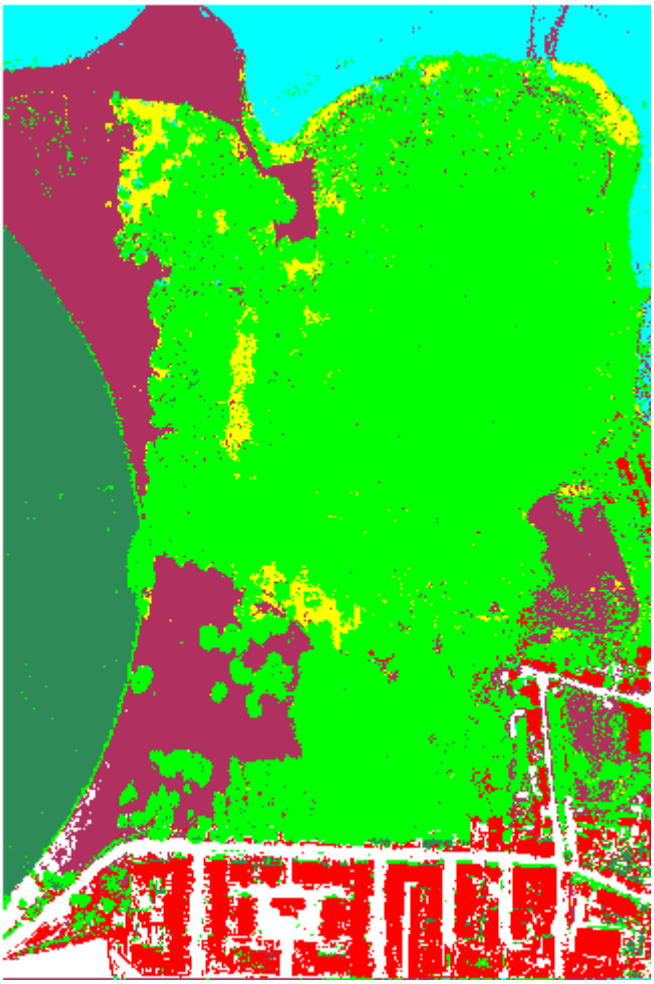
O ensaio descrito acima obteve uma precisão global no valor de 96,5861%, valor este superior ao mesmo ensaio que utilizou o algoritmo QUEST, no entanto os valores do atributo de Segmentação foram superiores.

Neste ensaio os valores obtidos para a segmentação foram muito menores, o que levou a uma imagem muito mais segmentada em termos de objectos a representar. Contudo, o algoritmo de geração de árvore de decisão aproveita este nível de detalhe para especificar mais regras que levam a diferenciar melhor as classes de ocupação do solo.

Por sua vez, os resultados obtidos da classificação foram bastante aceitáveis de acordo com a produção do utilizador. Mesmo a classe de mato, que previamente tinha tido uma classificação muito afastada dos valores de referência teve neste caso, valores de concordância acima dos 75%.

6. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	10	10	C4.5	97,3027%



P.G.: 97,3027% Kappa: 0.9600	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	182421	3	0	1345	0	151	1	183921	99.18	0.82
casas	0	32691	89	114	2	57	0	32953	99.20	0.80
estradas	4	98	20822	38	0	101	0	21063	98.86	1.14
floresta	1730	303	99	631779	8870	3731	1292	647804	97.53	2.47
mato	0	1	1	5852	28768	779	273	35674	80.64	19.36
solo a desc.	168	115	254	5428	1044	221009	306	228324	96.80	3.20
aq.lodo	0	0	10	1166	285	305	109558	111324	98.41	1.59
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	98.97	98.43	97.87	97.84	73.82	97.73	98.32			
E.O. (%)	1.03	1.57	2.13	2.16	26.18	2.27	1.68			

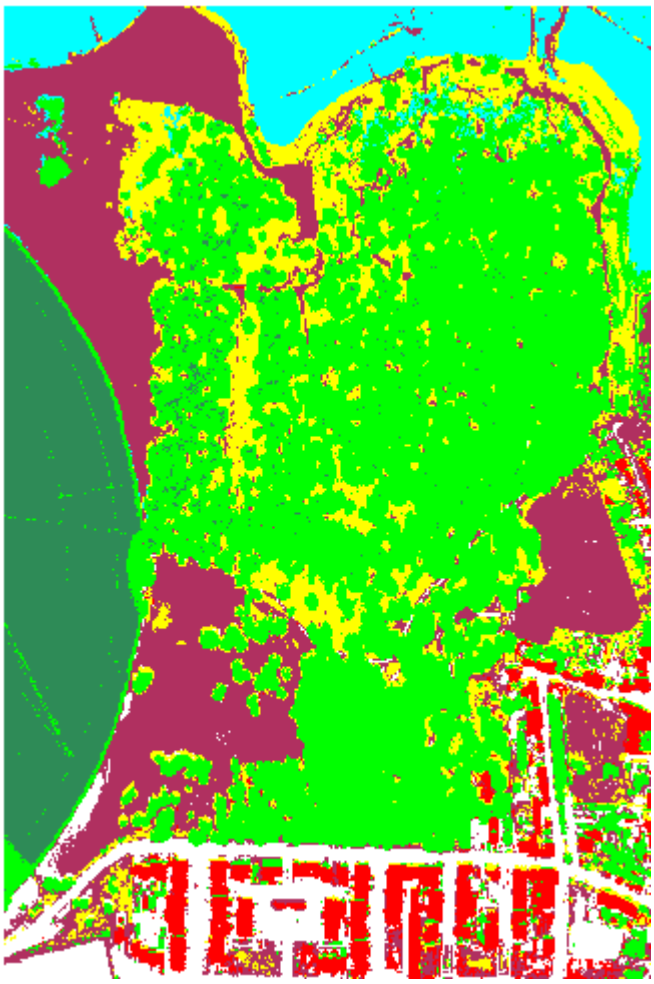
O algoritmo C4.5 mais uma vez demonstrou ser mais eficiente que o algoritmo QUEST, nos mesmos termos de comparação, ou seja, utilizando os mesmos atributos.

É de realçar que as duas classes, regadio e casas, na classificação obtida, atingiram quase 100% face aos dados de referência. O algoritmo conseguiu separar muito bem a classe regadio e a classe de casas, neste caso dentro do urbano.

Tanto a classe mato como a classe solo a descoberto confundem-se mais com a classe floresta, do que propriamente entre eles.

7. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	15	300	MAX Veros.	89.4001%

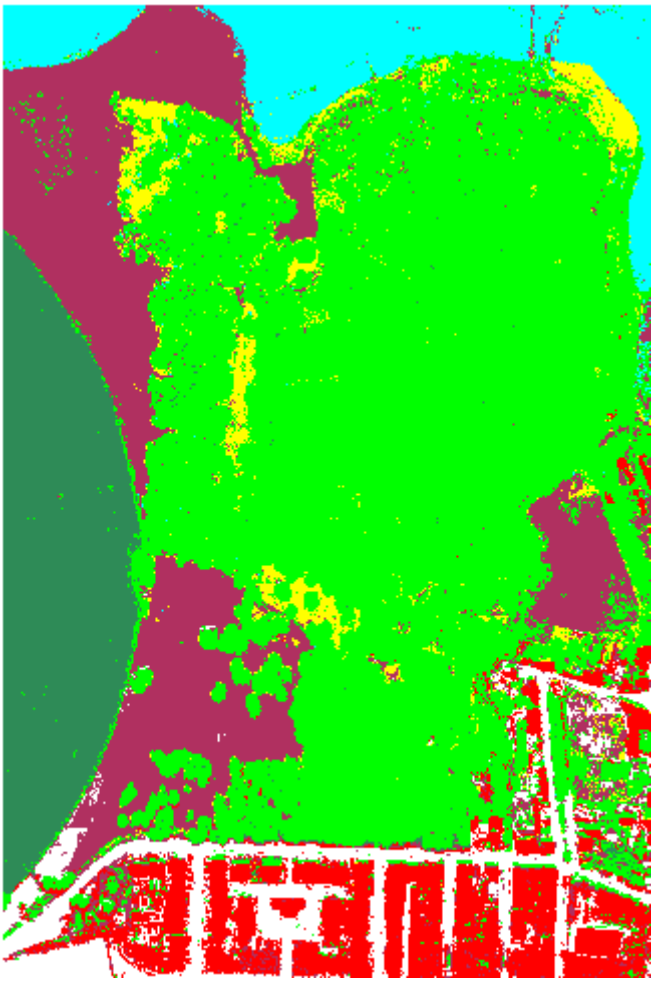


P.G.: 89.4001% Kappa: 0.8498	Terreno									
Classificação	**regadio**	**solo a desc.**	**aq.lodo**	**casas**	**estradas**	**floresta**	**mato**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	176174	18	0	0	0	5721	0	181913	96.85	4.42
solo a desc.	0	211496	1570	219	178	17542	2211	233216	90.69	6.47
aq.lodo	0	1095	106482	0	0	5775	273	113625	93.71	4.44
casas	0	132	0	32704	72	69	0	32977	99.17	1.53
estradas	0	1093	0	54	20909	643	1	22700	92.11	1.72
floresta	8149	6155	1190	75	23	544744	1602	561938	96.94	15.64
mato	0	6144	2188	159	93	71228	34882	114694	30.41	10.49
Total	184323	226133	111430	33211	21275	645722	38969	1261063		
P.P. (%)	95.58	93.53	95.56	98.47	98.28	84.36	89.51			
E.O. (%)	3.15	9.31	6.29	0.83	7.89	3.06	69.59			

A utilização de uma segmentação mais próxima dos objectos a classificar não é relevante para aumentar a precisão global do classificador. Muito embora, os resultados obtidos são satisfatórios, sendo este facto muito devido às áreas de treino utilizadas.

8. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
10%	15	300	QUEST univariado	97.0637%

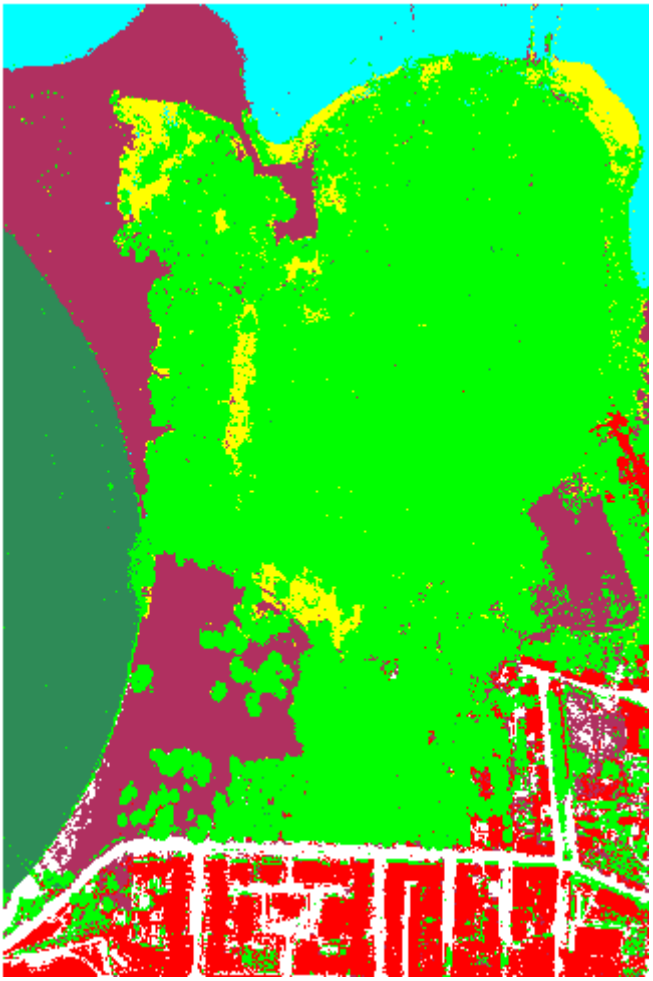


P.G.: 97,0637% Kappa: 0.9563	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	181369	0	0	1094	0	97	0	182560	99.35	0.65
casas	2	32698	103	124	1	76	0	33004	99.07	0.93
estradas	0	53	20906	10	0	174	0	21143	98.88	1.12
floresta	2592	270	59	632348	11644	3885	872	651670	97.04	2.96
mato	0	17	0	6021	26207	854	279	33378	78.52	21.48
solo a desc.	360	173	206	5249	946	220648	420	228002	96.77	3.23
aq.lodo	0	0	1	876	171	399	109859	111306	98.70	1.30
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	98.40	98.46	98.27	97.93	67.25	97.57	98.59			
E.O. (%)	1.60	1.54	1.73	2.07	32.75	2.43	1.41			

À medida que a segmentação se aproxima mais dos objectos reais que se pretendem classificar, melhores resultados apresentam os classificadores. No caso do algoritmo QUEST registou-se uma ligeira melhoria.

9. Execução

Áreas de treino (% das ROI de referência)	Segmentação		Algoritmo	Concordância
	Similaridade	Área (m2)		
10%	15	300	C4.5	98.1756%



P.G.: 98,1756% Kappa: 0.9729	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	182884	0	0	474	0	148	0	183506	99.66	0.34
casas	0	32809	98	62	1	25	0	32995	99.44	0.56
estradas	0	58	21054	56	0	74	0	21242	99.11	0.89
floresta	945	247	68	636227	6907	1890	547	646831	98.36	1.64

mato	0	0	0	4902	31280	375	146	36703	85.22	14.78
solo a desc.	494	97	55	3348	544	223395	330	228263	97.87	2.13
aq.lodo	0	0	0	653	237	226	110407	111523	99.00	1.00
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	99.22	98.79	98.96	98.53	80.27	98.79	99.08			
E.O. (%)	0.78	1.21	1.04	1.47	19.73	1.21	0.92			

E mais uma vez verifica-se que o algoritmo C4.5 também aumenta de qualidade quando a informação associada também aumenta, conseguindo também acompanhar a superioridade face ao algoritmo QUEST.

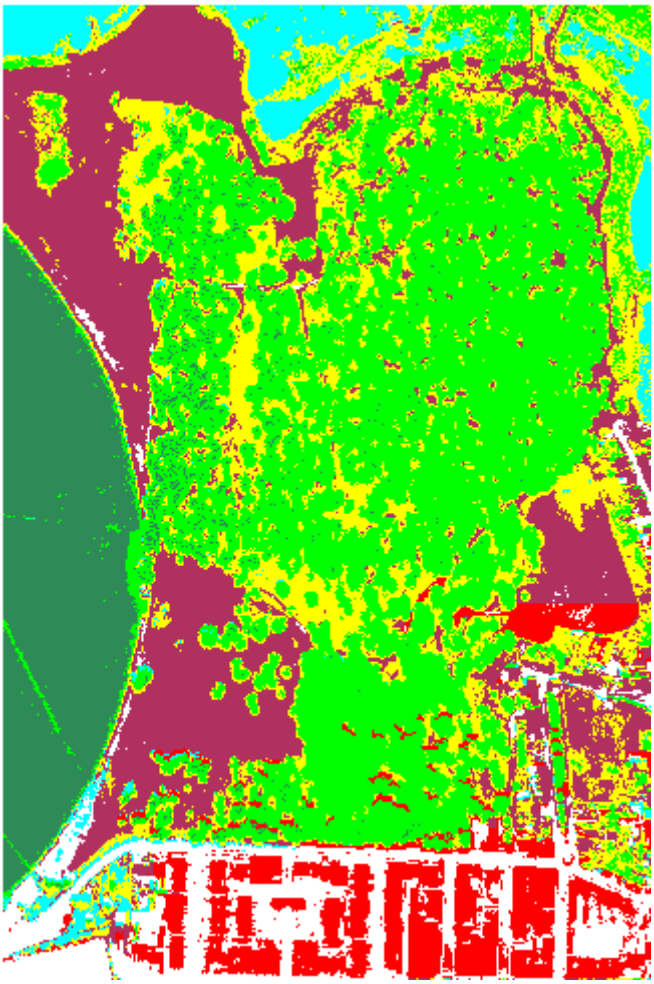
10. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	100	100	MAX. Veros.	Não obtido

O resultado obtido foi apenas uma classe preenchida para toda a imagem classificada.

11. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	100	100	QUEST univariado	81.5433%



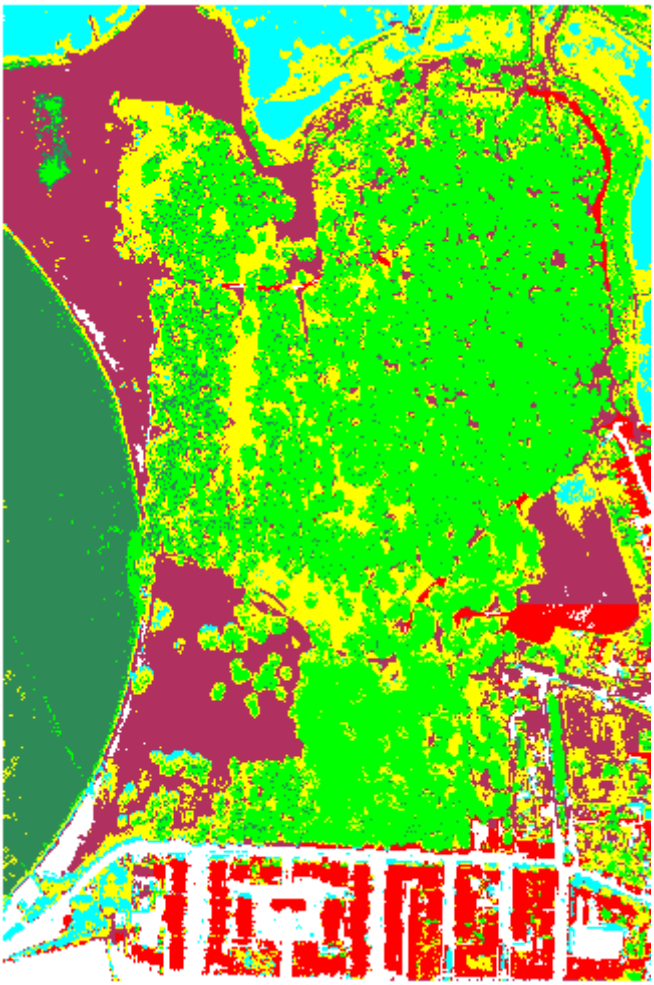
P.G.: 81,5433% Kappa: 0.7413	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	178263	0	0	9261	0	527	399	188450	94.59	5.41
casas	0	26301	329	3432	0	11	0	30073	87.46	12.54
estradas	0	258	19018	267	1	3651	36	23231	81.86	18.14
floresta	5896	1	69	510635	1748	4539	19594	542482	94.13	5.87
mato	101	304	35	94986	30701	11096	32862	170085	18.05	81.95
solo a desc.	0	6343	1426	26248	6269	205492	636	246414	83.39	16.61
aq.lodo	63	4	398	893	250	817	57903	60328	95.98	4.02
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	96.71	79.19	89.39	79.08	78.78	90.87	51.96			
E.O. (%)	3.29	20.81	10.61	20.92	21.22	9.13	48.04			

A classe mato foi a que obteve piores resultados de classificação. Os erros de comissão ultrapassaram os 80%, da qual maior parte foi classificada como floresta. Houve também confusão com a classe de aquicultura. Esta, por sua vez, foi a que teve

menor percentagem na precisão do produtor, ou seja, teve uma menor identificação na classificação relativamente aos seus dados de referência. A precisão global ronda os 80%, sendo uma boa precisão, contudo houve algumas confusões entre classes, nomeadamente terem sido identificados elementos urbanos, como as casas, nas classes de floresta ou mato.

12. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	100	100	C4.5	80.1450%

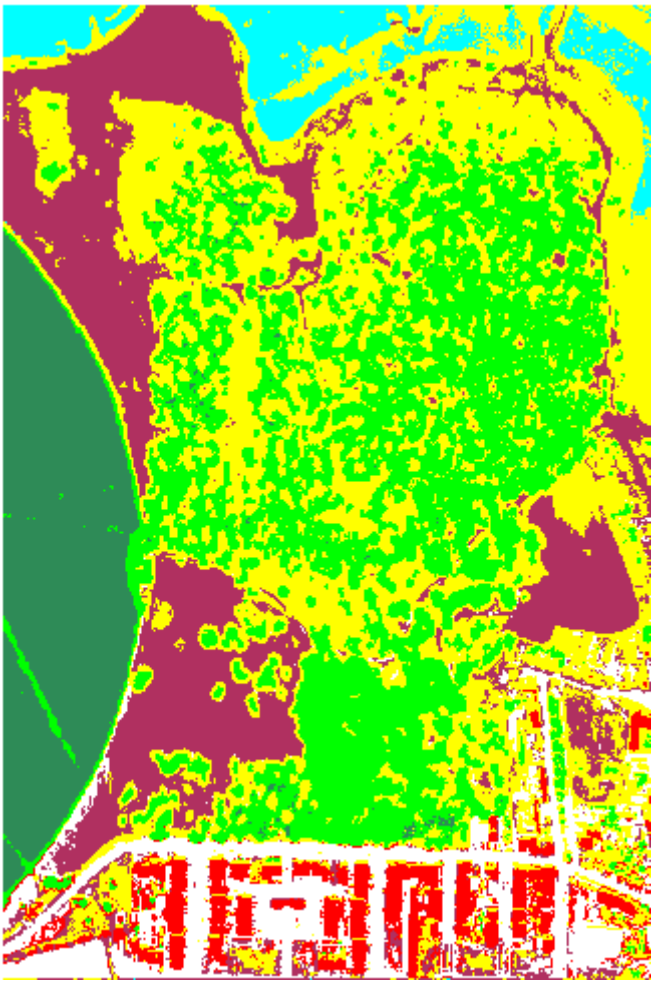


P.G.: 80,1450% Kappa: 0.7256	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	178625	0	2	26557	272	3771	2356	211583	82.65	17.35
Casas	0	26819	288	2962	7	0	0	30076	17.08	82.92
estradas	0	142	19194	190	0	3671	26	23223	84.42	15.58
floresta	3915	184	2	494543	2493	2677	8067	511881	96.61	3.39
Mato	1756	1526	85	94092	31316	10501	44073	183349	89.17	10.83
solo a desc.	2	4533	1431	26362	4523	204374	1100	242325	84.34	15.66
aq.lodo	25	7	273	1016	358	1139	55808	58626	95.19	4.81
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	90.22	80.36	96.91	76.59	80.75	90.38	50.08			
E.O. (%)	9.78	19.64	3.09	23.41	19.25	9.62	49.92			

A classe casas foi muito mal classificada neste ensaio. Foi identificada em outras classes erradamente, onde a classe floresta teve o maior peso. A classe de aquicultura continua a ter resultados de precisão de produção na ordem dos 50%, o que revela ser uma classe que se confunde com outras classes como mato e floresta. A precisão global também é aceitável, contudo a classe aquicultura muito afectou para degradar a classificação da imagem.

13. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	10	10	MAX Veros.	73.2865%



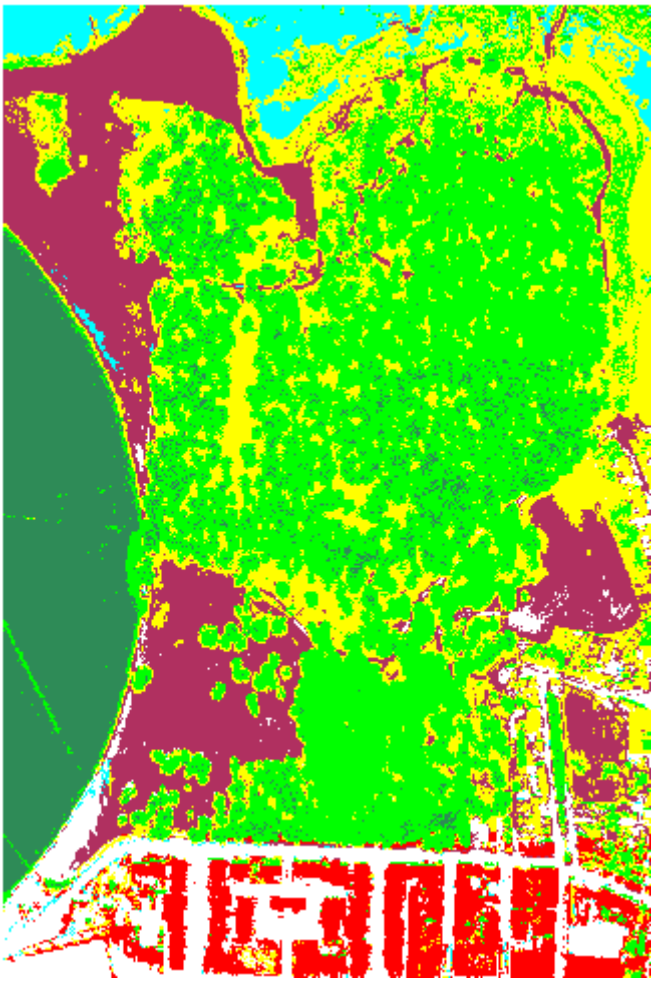
P.G.: 73,2865% Kappa: 0.6533	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	176511	0	0	2560	0	0	0	179071	98.57	1.43
casas	10	28704	46	107	0	0	0	28867	99.44	0.56
estradas	0	121	20724	361	1	1066	0	22273	93.05	6.95
floresta	7645	0	0	392963	300	2230	169	403307	97.44	2.56
mato	157	3724	150	237969	35402	21859	41997	341258	10.37	89.63
solo a desc.	0	662	355	11710	3266	200953	332	217278	92.49	7.51
aq.lodo	0	0	0	52	0	25	68932	69009	99.89	0.11
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	95.76	86.43	97.41	60.86	90.85	88.86	61.86			
E.O. (%)	4.24	13.57	2.59	39.14	9.15	11.14	38.14			

A classe floresta, como se pode visualizar claramente na imagem anterior foi a classe que teve o menor nível de precisão do produtor, ou seja, foi a classe menos identificada na classificação comparativamente com os dados de referência. A classe

mato obteve uma classificação muito errada dos dados de referência, pois teve resultados que neste caro seriam de outras classes. Este facto poderá dever-se a ter uma imagem muito segmentada, ou seja, o atributo de Similaridade ser de 10 e o de Área ser de 10 pixels. A precisão global é razoável, por estar acima dos 70% e o índice kappa ter um valor aproximadamente de 0,7.

14. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	10	10	QUEST univariado	80.3273%



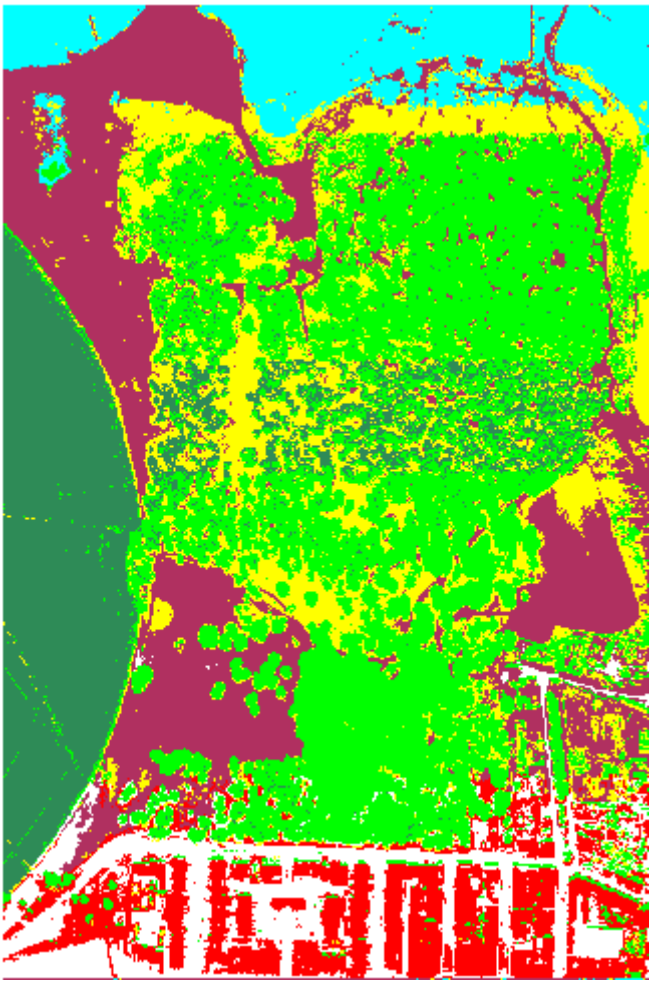
P.G.: 80,3273% Kappa: 0.7268	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	178628	0	0	24868	0	503	399	204398	87.39	12.61
Casas	0	26774	341	278	0	0	0	27393	97.74	2.26
estradas	31	609	20243	672	4	4597	0	26156	77.39	22.61
floresta	5573	17	40	506657	2275	4932	13874	533368	94.99	5.01
Mato	91	5340	126	101973	34551	17932	46778	206791	16.71	83.29
solo a desc.	0	470	444	11225	2071	196466	720	211396	92.94	7.06
aq.lodo	0	1	81	49	68	1703	49659	51561	96.31	3.69
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	96.91	80.62	95.15	78.46	88.66	86.88	44.57			
E.O. (%)	3.09	19.38	4.85	21.54	11.34	13.12	55.43			

A classe mato continua a ser a classe que tem piores resultados de classificação, e continua a ser distribuída pelas classes de floresta e aquicultura. A classe aquicultura,

por sua vez, foi a classe que teve piores resultados referentes aos dados de referência. A classificação identificou um valor para a classe de aquicultura inferior em 50% da área identificada como referência.

15. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	10	10	C4.5	79.7975%

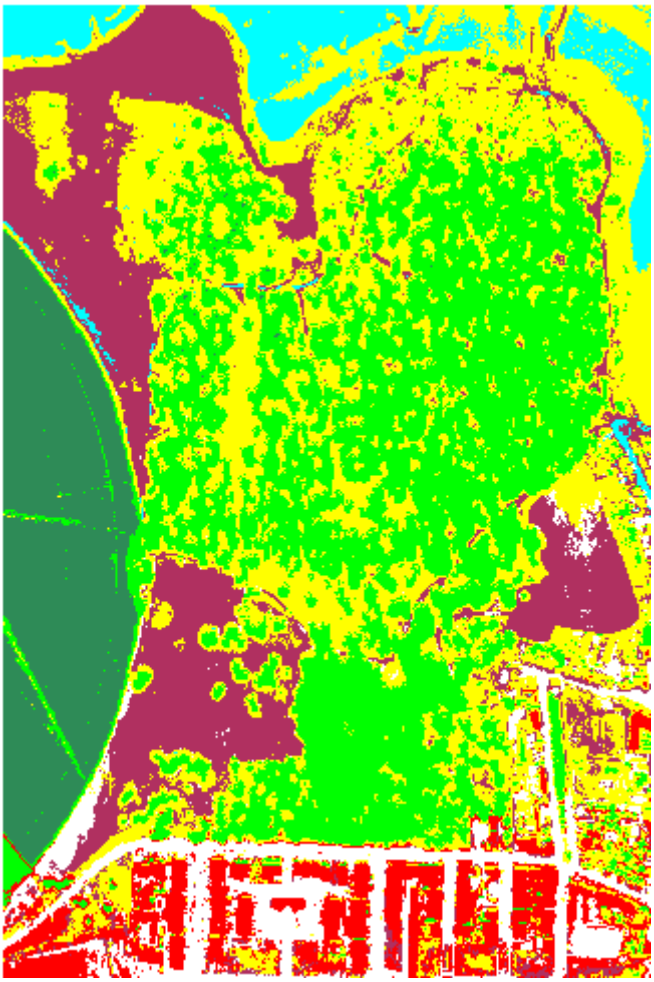


P.G.: 79,7975% Kappa: 0.7274	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	178798	0	0	50877	7	866	0	230548	77.55	22.45
Casas	5	27385	560	464	0	2185	0	30599	89.50	10.50
estradas	86	379	19860	1249	0	803	0	22377	88.75	11.25
floresta	4916	140	167	441541	2598	2701	1073	453136	97.44	2.56
Mato	516	513	38	111820	30761	9003	7803	160454	19.17	80.83
solo a desc.	2	4794	650	28583	4525	206500	1102	246156	83.89	16.11
aq.lodo	0	0	0	11188	1078	4075	101452	117793	86.13	13.87
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	97.00	82.46	93.35	68.38	78.94	91.32	91.05			
E.O. (%)	3.00	17.54	6.65	31.62	21.06	8.68	8.95			

A classe floresta foi a classe menos identificada comparativamente com os dados de referência e a classe regadio foi a classe que mais prejudicou este valor. A própria classe mato foi muito mal classificada, onde obteve maior parte dos dados que deveriam pertencer à classe de floresta. As classes estrada e aquicultura foram as classes que melhores resultados apresentaram, em termos de precisão de produtor como de precisão de utilizador.

16. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	15	300	MAX Veros.	76.5955%



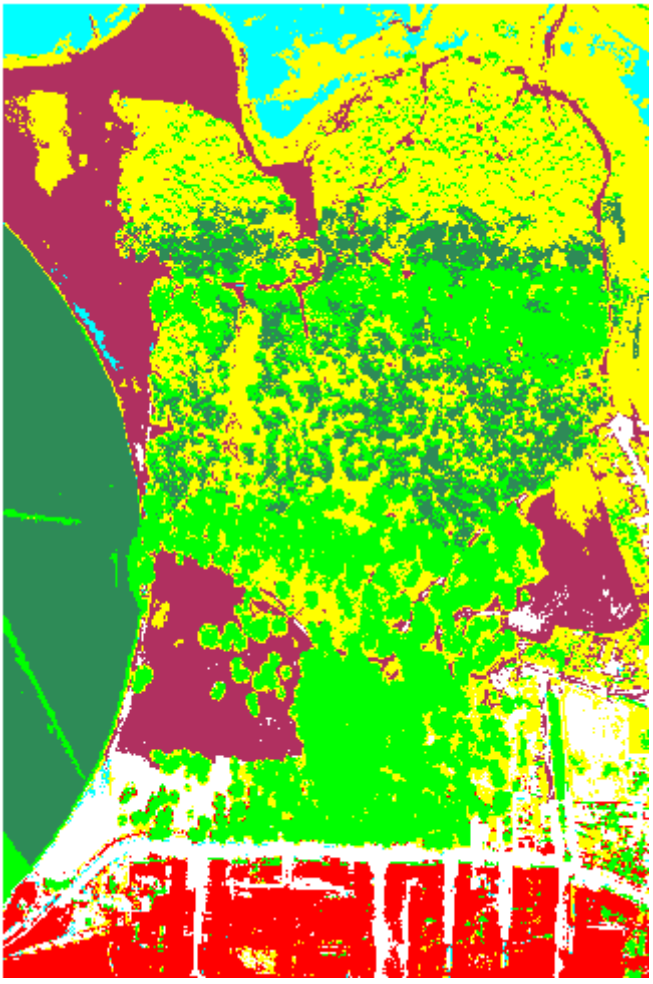
P.G.: 76,5955% Kappa: 0.6892	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	171795	0	0	441	0	14	0	172250	99.74	0.26
Casas	391	28817	265	238	0	0	0	29711	96.99	3.01
estradas	0	46	20718	460	12	2058	0	23294	88.94	11.06
floresta	11322	26	0	438474	336	1295	77	451530	97.11	2.89
Mato	809	3261	213	194180	36658	24311	37982	297414	12.33	87.67
solo a desc.	0	1059	79	11466	1960	196384	300	211248	92.96	7.04
aq.lodo	6	2	0	463	3	2071	73071	75616	96.63	3.37
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	93.20	86.77	97.38	67.90	94.07	86.84	65.58			
E.O. (%)	6.80	13.23	2.62	32.10	5.93	13.16	34.42			

A classe mato foi quase toda classificada, contudo foi também classificada em áreas onde outras classes deveriam ter sido classificadas (produção de utilizador baixa).

As classes mais agregadas pela classe mato foram a floresta e a aquicultura. As classes de urbano, como casas e estradas tiveram valores muito aceitáveis e por análise e visualização da imagem classificada estes foram bem distribuídos.

17. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	15	300	QUEST Univariado	63.8956%

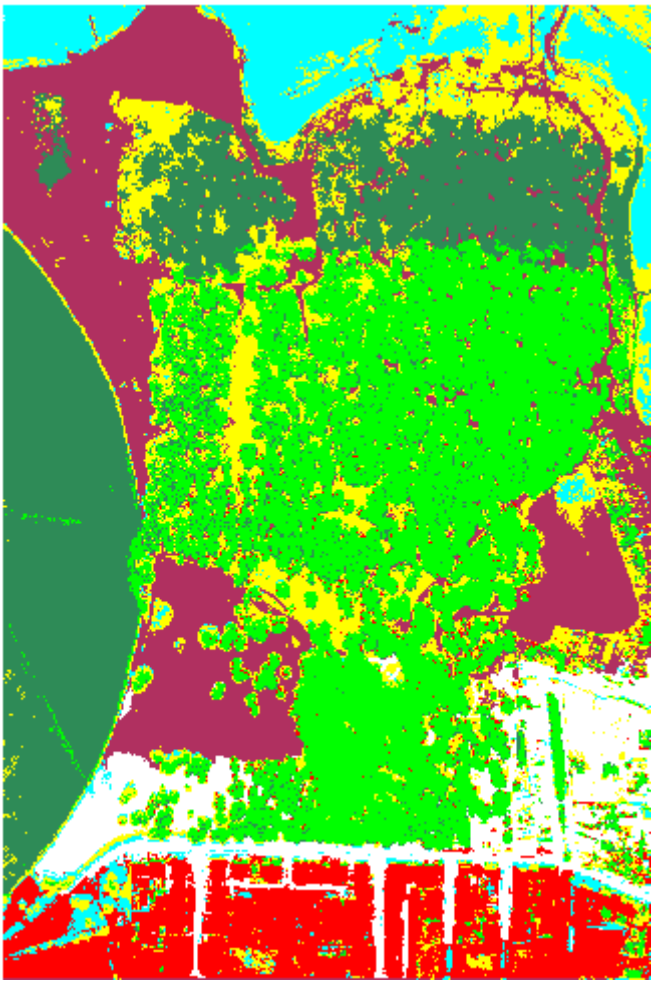


P.G.: 63,8956% Kappa: 0.5439	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	175302	0	0	102108	545	2126	0	280081	62.59	37.41
Casas	0	26326	2790	463	0	14	0	29593	88.96	11.04
estradas	31	937	17217	1197	4	19196	0	38582	44.62	55.38
floresta	8947	18	38	318981	1537	500	899	330920	96.39	3.61
Mato	42	5554	153	209551	34185	18638	59958	328081	10.42	89.58
solo a desc.	1	375	996	13364	2624	183900	720	201980	91.05	8.95
aq.lodo	0	1	81	58	74	1759	49853	51826	96.19	3.81
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	95.11	79.27	80.93	49.40	87.72	81.32	44.74			
E.O. (%)	4.89	20.73	19.07	50.60	12.28	18.68	55.26			

A classificação resultante em termos de precisão global e de índice kappa obteve resultados médios. A classe mato e estradas foram muito mal classificadas, e as classes floresta e aquicultura foram muito pouco identificadas relativamente aos seus dados de referência.

18. Execução

Áreas de treino (% das ROI de referência)	**Segmentação**		**Algoritmo**	**Concordância**
	Similaridade	**Área (m2)**		
ROI manuais	15	300	C4.5	73.7201%



P.G.: 79,7975% Kappa: 0.7274	Terreno									
Classificação	**regadio**	**casas**	**estradas**	**floresta**	**mato**	**solo a desc.**	**aq.lodo**	**Total**	**P.U. (%)**	**E.C. (%)**
regadio	180296	6	2	137296	2708	5509	290	326107	55.29	44.71
casas	0	25762	4140	4516	35	1	0	34454	74.77	25.23
estradas	0	5487	16805	891	0	17957	24	41164	40.82	59.18
floresta	2302	189	15	395841	981	77	1	399406	99.11	0.89
mato	1704	606	55	77317	29691	7848	21841	139062	21.35	78.65
solo a desc.	2	1154	7	28030	4525	193090	1102	227910	84.72	15.28
aq.lodo	19	7	251	1831	1029	1651	88172	92960	94.85	5.15
Total	184323	33211	21275	645722	38969	226133	111430	1261063		
P.P. (%)	97.82	77.57	78.99	61.30	76.19	85.39	79.13			
E.O. (%)	2.18	22.43	21.01	38.70	23.81	14.61	20.87			

As classes de estradas e de mato foram as classes que piores classificações obtiveram. A classe de floresta apesar de ter tido uma produção de utilizador de quase 100%, teve um valor de precisão de produtor muito baixa, aproximadamente de 60%.

Houve de facto uma grande confusão com a classe regadio, principalmente em zonas que deveriam ter sido consideradas como floresta. A segmentação que mais se aproximava dos objectos reais, não produziu os resultados esperados, principalmente na precisão de produtor.

IV.6 Concordância dos Resultados

A concordância dos resultados sintetiza os resultados obtidos no subcapítulo anterior, comparando cada ensaio relativamente à precisão global de cada um.

Execução	**Áreas de treino (% das ROI de referência)**	**Segmentação**		**Algoritmo**	**Concordância**
		Similaridade	**Área (m2)**		
1.	10%	100	100	MAX Veros.	Não obtido
2.	10%	100	100	QUEST univariado	93,2241%
3.	10%	100	100	C4.5	93.5336%
4.	10%	10	10	MAX Veros.	89,3873%
5.	10%	10	10	QUEST univariado	96.5861%
6.	10%	10	10	C4.5	97,3027%
7.	10%	15	300	MAX Veros.	89,4001%
8.	10%	15	300	QUEST univariado	97.0637%
9.	10%	15	300	C4.5	98.1756%

10.	ROI manuais	100	100	MAX Veros.	Não obtido
11.	ROI manuais	100	100	QUEST univariado	81.5433%
12.	ROI manuais	100	100	C4.5	80.1450%
13.	ROI manuais	10	10	MAX Veros.	73.2865%
14.	ROI manuais	10	10	QUEST univariado	80.3273%
15.	ROI manuais	10	10	C4.5	79.7975%
16.	ROI manuais	15	300	MAX Veros.	76.5955%
17.	ROI manuais	15	300	QUEST univariado	63.8956%
18.	ROI manuais	15	300	C4.5	73.7201%
19.	ROI manuais	15	300	QUEST linear	75.3861%
20.	ROI manuais	-	-	MAX Veros.	75.5964%
21.	ROI manuais	-	-	QUEST univariado	79.6692%
22.	ROI manuais	-	-	C4.5	79.1762%

A tabela anterior demonstra que para as áreas de treino aleatórias, o algoritmo C4.5 foi o algoritmo que melhores resultados obteve face ao algoritmo QUEST. No entanto os valores da precisão global são muito próximos entre eles. Já para as áreas de treino manuais, o algoritmo QUEST na maior parte dos ensaios obteve resultados melhores.

A variável de segmentação utilizada como atributo de classe no conjunto de variáveis independentes para a geração da árvore de decisão também demonstrou ser um factor relevante na precisão global do classificador, ou seja, quanto mais precisa é a segmentação ajustada aos objectos reais que se pretendem classificar, melhores resultados se obtêm, neste caso para as áreas de treino aleatórias. No caso das áreas de treino manuais, onde foram aplicados a mesma ordem de ensaios das áreas de treino aleatórias, os resultados de segmentação não obtiveram melhores resultados para o caso em que os segmentos se aproximam dos objectos reais e o próprio algoritmo QUEST foi o que apresentou piores resultados.

De salientar também que a confusão espectral aparente entre as classes de floresta e regadio foram muito pouco apresentadas na áreas de treino aleatórias, principalmente devido à técnica de segmentação. Já as áreas de treino manuais, revelaram uma confusão espectral maior entre as classes, para vários níveis de segmentação, nomeadamente entre floresta, mato e aquicultura.

Os resultados dos ensaios de combinação linear no algoritmo QUEST não foram apresentados, pois estes em termos de precisão global revelaram-se inferiores, ou seja, para o ensaio de segmentação com os valores de Similaridade de 15 e da Área com 300 pixels, foi exactamente de 96.3693%. Contudo para o mesmo ensaio, no entanto com as áreas de treino manuais, procedeu-se à execução do algoritmo QUEST por combinação linear e este apresentou uma precisão global de 75.3861%, quase 10% superior face ao QUEST univariado.

É importante referir que a imagem continha sombra derivada das árvores e das casas, no entanto esta não foi considerada. Por esta razão deve-se assumir uma percentagem de erro aos valores da classificação das classes.

Para tentar perceber os valores baixos dos ensaios com as áreas de treino manuais e os valores de segmentação mais próximos dos objectos reais, excluiu-se a variável de segmentação na execução da classificação a partir das árvores de decisão com o objectivo de verificar se esta tinha influência na precisão global. A exclusão da variável veio a revelar-se determinante para obter uma melhor precisão global, e o algoritmo QUEST foi o que obteve melhores resultados, comparando entre o método de máxima verosimilhança e o algoritmo C4.5.

Os resultados obtidos, em geral, resultam de uma variabilidade espectral inerente de imagens de grande resolução espacial, mas há situações em que essa variabilidade ainda é maior, como é o caso dos resultados obtidos com as áreas de treino manuais.

As árvores de decisão geradas pelos algoritmos QUEST e o C4.5 obtiveram resultados aceitáveis e na maioria melhores quando comparados com o método de máxima verosimilhança. Contudo, a conclusão dos ensaios realizados levam a que poderá haver situações em que uma das variáveis de atributos usadas para gerar a árvore de decisão tanto no algoritmo QUEST e C4.5 poderá ser determinante para obter resultados piores. Por vezes o acrescento de informação em termos de variáveis para a geração da árvore de decisão, nem sempre melhora a precisão da classificação. Neste caso é importante proceder a técnicas de validação e incrementar os seus valores para garantir uma árvore de decisão precisa à classificação que se pretende alcançar.

Capitulo V: Conclusões

As árvores de decisão começam a ser uma possibilidade real para as novas situações de classificação, principalmente perante a evolução das imagens de satélite que têm cada vez mais uma maior resolução e onde se pretende colmatar problemas comuns na área da detecção remota.

As árvores de decisão demonstraram ser rápidas, eficientes e não requerem conhecimento aprofundado para a sua geração e execução de classificação de imagens.

Os algoritmos QUEST e C4.5 por serem algoritmos de geração de árvores de decisão univariadas demonstraram conseguir resolver em tempo útil os problemas de classificação de imagens de muito alta resolução. Cada um tem características diferentes, contudo, dada a sua natureza, os resultados apresentados foram muito precisos.

Os ensaios com o método de máxima verosimilhança demonstraram que a classificação a partir de árvores de decisão obteve em geral resultados melhores. Contudo foi verificado no ensaio, onde a segmentação se aproximava mais dos objectos reais, que os resultados foram melhores para o método de máxima verosimilhança.

De forma a comparar os resultados de classificação de ambos os algoritmos, foi desenvolvida uma interface de programação que permitiu integrar diversos programas de software de modo a garantir a interoperabilidade entre as aplicações. Esta veio a revelar-se de grande utilidade e eficiência nos resultados obtidos.

Uma das técnicas apresentada na metodologia proposta, designadamente a segmentação, comprovou em geral ser uma vantagem para a identificação dos objectos de classe através de segmentos gerados, servindo como base do conjunto de atributos que ajudariam a criar as regras de classificação para árvore de decisão.

Foi importante verificar que a análise realizada ao processo de segmentação sobre a aproximação da forma dos segmentos aos objectos reais da imagem a classificar, foram relevantes para a precisão global das várias classificações, independentemente do algoritmo utilizado, neste caso para as áreas de treino aleatórias. Já para as áreas de treino manuais, o mesmo aconteceu, à excepção de um dos ensaios, o que mais aproximava os segmentos dos objectos reais, que obteve resultados pouco aceitáveis. Este facto foi confrontado com novos ensaios, mas excluindo a segmentação. Os

resultados continuaram a favorecer a classificação produzida pelas árvores de decisão e neste caso o algoritmo QUEST foi o que obteve melhores resultados. Uma das hipóteses, para tal facto, poderá estar relacionada com o problema de sobre-aprendizagem nos algoritmos de geração de árvores de decisão. No entanto deverá ser alvo de verificação em trabalhos futuros.

Os atributos de classe escolhidos foram considerados que mais faziam sentido para o tipo de classificação que se pretendia obter. Deste conjunto de atributos, a escolha do atributo NDVI e o do atributo de segmentação, serão os atributos com maior peso nas inúmeras regras criadas para a geração da árvore de decisão.

Por um lado o NDVI é uma variável que avalia muito bem a existência de vegetação e por outro, o facto de dividir a imagem em segmentos permite a caracterização de objectos, essencialmente encontrados em meio urbano.

Uma vez que as classes resultantes eram constituídas tanto por classes de urbano, nomeadamente casas e estradas, como por classes de ocupação do solo, tais como floresta e solo a descoberto, a escolha destes atributos foram fundamentais. Contudo, a área de estudo tinha pouca diversidade de ocupação de solo.

Por outro lado, a utilização de mais atributos poderia melhorar a classificação pois seriam mais variáveis a distinguir as classes durante a geração da árvore de decisão. Ou seja, poderia permitir uma maior discriminação das classes que mais se aproximam da realidade. No entanto, deve-se sempre avaliar as classes que se pretendem classificar e as características das imagens para determinar os melhores atributos que se poderão utilizar.

Os resultados obtidos, em geral, resultam de uma variabilidade espectral inerente de imagens de grande resolução espacial, mas há situações em que essa variabilidade ainda é maior, como é o caso dos resultados obtidos com as áreas de treino manuais.

É importante realizar mais ensaios sobre áreas mais diversas espectralmente para confrontação com os tão bons resultados de concordância obtidos.

A segmentação através do programa SPRING revelou obter resultados satisfatórios, visto que os segmentos tentam aproximar-se dos objectos reais presentes na imagem, estes poderiam agregar mais atributos que pudessem ajudar na classificação. Ou seja, podendo-se atribuir a cada um dos segmentos novos atributos de classificação, próprios dos segmentos, tais como área, perímetro ou outros relevantes

para os objectivos pretendidos, ganhava-se mais informação que poderia ajudar no processo de classificação.

Novas metodologias que vão surgindo procuram combinar métodos já existentes para aproveitar as vantagens de cada um e tentar minimizar as suas próprias limitações.

A metodologia proposta na presente dissertação resultou de uma combinação de metodologias conhecidas na área de detecção remota, complementada com a componente prática de as juntar experimentando novos métodos com recurso a novas aplicações.

Esta é sem dúvida uma possível abordagem a adoptar no futuro, para o desenvolvimento de produtos orientados para a resolução de diversas situações que a comunidade da detecção remota vai enfrentando com os novos recursos que vão surgindo e com as novas ferramentas que são desenvolvidas.

A metodologia Shackelford e Davis apresentada na presente dissertação como caso de estudo utilizando as árvores de decisão, decorre desta linha de orientação e começa a servir de base para outras investigações que voltam a enfrentar novas situações derivadas de métodos que foram considerados no passado muito eficientes mas que com a evolução dos dados e dos recursos disponíveis começam a revelar as suas insuficiências.

Estas novas metodologias têm um potencial interesse para a comunidade da detecção remota envolvida no desenvolvimento de sistemas operacionais baseados em imagens de satélite, nomeadamente os previstos pelos programas internacionais a decorrer, tais com o GEOSS e o GMES.

A presente tese procurou assim apresentar também novas abordagens de estudos recentes que poderão vir a contribuir, na prática, para a melhoria dos sistemas actualmente desenvolvidos e deixar em aberto novos ensaios e testes com o objectivo último de melhorar sempre o resultado final obtido.

BIBLIOGRAFIA(S) / REFERÊNCIAS BIBLIOGRÁFICAS

Abdelhamid A. Elnaggar e Jay S. Noller, 2010, Application of Remote-sensing Data and Decision-Tree Analysis to Mapping Salt-Affected Soils over Large Areas, Remote Sens. 2010, 2, 151-165; doi:10.3390/rs2010151, ISSN 2072-4292

Baraldi, A., Durieux, L., Simonetti, D., Conchedda, G., Holecz, F., Blonda, P., 2010, Automatic Spectral-Rule-Based Preliminary Classification of Radiometrically Calibrated SPOT-4/-5/IRS, AVHRR/MSG, AATSR, IKONOS/QuickBird/OrbView/GeoEye, and DMC/SPOT-1/-2 Imagery—Part I: System Design and Implementation, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 48, NO. 3, MARCH 2010

Baraldi, A., Durieux, L., Simonetti, D., Conchedda, G., Holecz, F., Blonda, P., 2010, Automatic Spectral Rule-Based Preliminary Classification of Radiometrically Calibrated SPOT-4/-5/IRS, AVHRR/MSG, AATSR, IKONOS/QuickBird/OrbView/GeoEye, and DMC/SPOT-1/-2 Imagery—Part II: Classification Accuracy Assessment, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 48, NO. 3, MARCH 2010

BENZ, U., HOFMANN, P., WILLHAUCK, G., LINGENFELDER, I. and HEYNEN, M., 2004, Multiresolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. ISPRS Journal of Photogrammetry and Remote Sensing, 58, pp. 239–258.

Bouziani, M., Goita, K., e He.,D., 2010, Rule-Based Classification of a Very High Resolution Image in an Urban Environment Using Multispectral Segmentation Guided by Cartographic Data, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 48, NO. 8, AUGUST 2010

Breiman, L., Friedman, J.H., Olshen, R., e Stone, C.J., 1984. Classification and Regression Tree Wadsworth & Brooks/Cole Advanced Books & Software, Pacific California.

Carleer, A. P., Debeir, O. e Wolff, E., Assessment of very high spatial resolution satellite image segmentations, 2005, Photogramm. Eng. Remote Sens., vol. 71, no. 11, pp. 1285–1294, Nov. 2005.

Congalton, R. G. and Green, K., 1999, Assessing the Accuracy of Remotely Sensed Data. Boca Raton, FL: Lewis Publishers, 1999.

Colton, Simon, 2004. "Lecture 11 – Decision Tree Learning", Disponivel em: http://www.doc.ic.ac.uk/~sgc/teaching/v231/lecture11.html (visitado em 12 de Janeiro de 2011)

Foody, G. M., 2002, "Status of land cover classification accuracy assessment," Remote Sens. Environ., vol. 80, no. 1, pp. 185–201, Apr. 2002.

Friedl, M. e Brodley, C. 1997, Decision tree classification of land cover from remotely sensed data. Remote Sensing of Environment 61: 399-409

FROHN, R.C., HINKEL, K.M. and EISNER, W.R., 2005, Satellite remote sensing classification of thaw lakes and drained thaw lake basins on the North Slope of Alaska. Remote Sensing of Environment, 97, pp. 116–126.

Haralick, R., e Shapiro, L. (1985). Survey: image segmentation techniques. Computer Vision, Graphics, and Image Processing, 29, 100-132.

HODGSON, M.E., JENSEN, J.R., TULLIS, J.A., RIORDAN, K.D. and ARCHER, C.M., 2003, Synergistic use of lidar and color aerial photography for mapping urban parcel imperviousness. Photogrammetric Engineering and Remote Sensing, 69, pp. 973–980.

Homer, C., Huang, C., Yang, L., Wylle, B. & Coan, M. 2004. Development of a 2001 National Land-Cover Database for the United States. Photogrammetric engineering & Remote Sensing, 70 (7): 829-840.

HUANG, X. and JENSEN, J.R., 1997, A machine-learning approach to automated knowledgebase building for remote sensing image analysis with GIS data. Photgrammetric Engineering and Remote Sensing, 63, pp. 1185–1194.

JENSEN, J.R., 2005, Introductory Digital Image Processing: A Remote Sensing Perspective 3rd (Upper Saddle River, NJ: Prentice-Hall).

JENSEN, J.R., QUIJANO, M., HADLEY, B., IM, J., WANG, Z. and NEL, A.L., et al., 2006, Remote sensing agricultural crop type for sustainable development in South Africa. Geocarto International, 21, pp. 5–18.

Jensen, J. R. e Tullis, J. A., 2008, Object-based change detection using correlation image analysis and image segmentation, International Journal of Remote Sensing, 29: 2, 399 — 423, First published on: 09 June 2007 (iFirst)

LALIBERTE, A.S., RANGO, A., HAVSTAD, K.M., PARIS, J.F., BECK, R.F. and MCNEELY, R., et al., 2004, Object-oriented image analysis for mapping shrub encroachment from 1937 to 2003 in southern New Mexico. Remote Sensing of Environment, 93, pp. 198–210.

Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees, *Statistica Sinica* **7**: 815– 840.

MUCHONEY, D., GOPAL, S., HODGES, J., MORROW, N., STRAHLER, A. and BORAK, J., et al., 2000, Application of the MODIS global supervised classification model to vegetation and land cover mapping of central America. International Journal of Remote Sensing, 21, pp. 1115–1138.

Pal, M. & Mather, M. 2003. An assessment of the effectiveness of decision tree methods for land cover classification. Remote Sensing of Environment 86: 554-565.

QUINLAN, J.R., 2003, Data Mining Tools See5 and C5.0, St. Ives NSW, Australia: RuleQuest Research. Available online at: http://www.rulequest.com/see5-info.html (accessed 21 January 2005).

Quinlan J., 1993, C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993. San Mateo, CA.

Rogan, J., Franklin, J. & Roberts, A. 2002. A comparison of methods for monitoring multitemporal vegetation change using Thematic Mapper imagery. Remote Sensing of Environment 80: 143– 156.

RULEQUEST RESEARCH, 2005, C5.0 Release 2.01 (data mining software) (St. Ives, Australia: RuleQuest Research).

Shackelford, A. K. e Davis,C. H., 2003, “Fully automated road network extraction from high-resolution satellite multispectral imagery,” in Proc. IGARSS, Toulouse, France, Jul. 2003, vol. 1, pp. 461–463.

Shackelford, A. K. e Davis,C. H., 2003, “A hierarchical fuzzy classification approach for high-resolution multispectral data over urban areas,” IEEE Trans. Geosci. Remote Sens., vol. 41, no. 9, pp. 1920–1932, Sep. 2003.

Shackelford, A. K. e Davis,C. H., 2003, "A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas," IEEE Trans. Geosci. Remote Sens., vol. 41, no. 10, pp. 2354–2363, Oct. 2003.

Sorel, Luc, 2010, WekaText2Xml, Disponivel em: http://www.lucsorel.com/index.php?page=downloads#wekatext2xml

Sreerama K. Murthy, 1998. "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey", Data Mining and Knowledge Discovery, 2, 345–389 (1998), 1998 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Teknomo, Kardi, 2009. "Tutorial on Decision Tree", Disponível em: http://people.revoledu.com/kardi/tutorial/DecisionTree/ (visitado em 12 de Janeiro de 2011)

Xu, M., Watanachaturaporn, P., Varshney, P. K. & Arora, M. 2005. Decision tree regression for soft classification of remote sensing data. Remote Sensing of Environment 97: 322-336.

Disponível em: http://wekadocs.com/node/2 (visitado em 18 de Novembro de 2010)

LISTA DE FIGURAS

LISTA DE TABELAS

ANEXOS

ANEXO I – Metadados da Fotografia Aérea Digital

```
<?xml version="1.0" encoding="UTF-8"?>

       <gmd:MD_Metadata                xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:gmd="http://www.isotc211.org/2005/gmd"       xmlns:gml="http://www.opengis.net/gml"
xmlns:smXML="http://metadata.dgiwg.org/smXML"  xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xs="http://www.w3.org/2001/XMLSchema"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"                            xsi:schemaLocation="http://www.isotc211.org/2005/gmd
http://www.isotc211.org/2005/gmd/gmd.xsd">

        <gmd:fileIdentifier>

            <gco:CharacterString>IGP_CGC_ORTO_004323BRGBI_07</gco:CharacterString>

        </gmd:fileIdentifier>

        <gmd:language>

            <gmd:LanguageCode                                    codeList="LanguageCode"
codeListValue="pt">pt</gmd:LanguageCode>

        </gmd:language>

        <gmd:hierarchyLevel>

            <gmd:MD_ScopeCode                                    codeList="#MD_ScopeCode"
codeListValue="tile">Folha</gmd:MD_ScopeCode>

        </gmd:hierarchyLevel>

        <gmd:contact>

            <gmd:CI_ResponsibleParty>

                <gmd:individualName>

                    <gco:CharacterString>DIRECÇÃO DE SERVIÇOS DE GEODESIA E CARTOGRAFIA
(DSGC)</gco:CharacterString>

                </gmd:individualName>

                <gmd:organisationName>

                    <gco:CharacterString>INSTITUTO         GEOGRÁFICO         PORTUGUÊS
(IGP)</gco:CharacterString>

                </gmd:organisationName>

                <gmd:contactInfo>

                    <gmd:CI_Contact>

                        <gmd:phone>

                            <gmd:CI_Telephone>

                                <gmd:voice>
```

<gco:CharacterString>+351213819600</gco:CharacterString>
                                </gmd:voice>
                                <gmd:facsimile>
<gco:CharacterString>+351213819699</gco:CharacterString>
                                </gmd:facsimile>
                            </gmd:CI_Telephone>
                        </gmd:phone>
                        <gmd:address>
                            <gmd:CI_Address>
                                <gmd:deliveryPoint>
                                    <gco:CharacterString>RUA ARTILHARIA UM,
107</gco:CharacterString>
                                </gmd:deliveryPoint>
                                <gmd:city>
                                    <gco:CharacterString>LISBOA</gco:CharacterString>
                                </gmd:city>
                                <gmd:postalCode>
                                    <gco:CharacterString>1099-052
LISBOA</gco:CharacterString>
                                </gmd:postalCode>
                                <gmd:country>
                                    <gco:CharacterString>PORTUGAL</gco:CharacterString>
                                </gmd:country>
                                <gmd:electronicMailAddress>
<gco:CharacterString>dsgc@igeo.pt</gco:CharacterString>
                                </gmd:electronicMailAddress>
                            </gmd:CI_Address>
                        </gmd:address>
                    </gmd:CI_Contact>
                </gmd:contactInfo>
                <gmd:role>
                    <gmd:CI_RoleCode codeList="#CI_RoleCode"
codeListValue="pointOfContact">contacto</gmd:CI_RoleCode>
                </gmd:role>
            </gmd:CI_ResponsibleParty>

```
        </gmd:contact>
        <gmd:dateStamp>
            <gco:Date>2008-01-31</gco:Date>
        </gmd:dateStamp>
        <gmd:referenceSystemInfo>
            <gmd:MD_ReferenceSystem>
                <gmd:referenceSystemIdentifier>
                    <gmd:RS_Identifier>
                        <gmd:code>
                            <gco:CharacterString>3763</gco:CharacterString>
                        </gmd:code>
                        <gmd:codeSpace>
                            <gco:CharacterString>EPSG</gco:CharacterString>
                        </gmd:codeSpace>
                    </gmd:RS_Identifier>
                </gmd:referenceSystemIdentifier>
            </gmd:MD_ReferenceSystem>
        </gmd:referenceSystemInfo>
        <gmd:identificationInfo>
            <gmd:MD_DataIdentification>
                <gmd:citation>
                    <gmd:CI_Citation>
                        <gmd:title>
                            <gco:CharacterString>Ortofotocarta                          IGP
004323B</gco:CharacterString>
                        </gmd:title>
                        <gmd:alternateTitle>
                            <gco:CharacterString>004323B</gco:CharacterString>
                        </gmd:alternateTitle>
                        <gmd:date>
                            <gmd:CI_Date>
                                <gmd:date>
                                    <gco:Date>2007-08-01</gco:Date>
                                </gmd:date>
                                <gmd:dateType>
```

<gmd:CI_DateTypeCode codeList="#CI_DateTypeCode" codeListValue="creation">criação</gmd:CI_DateTypeCode>

</gmd:dateType>

</gmd:CI_Date>

</gmd:date>

<gmd:edition>

<gco:CharacterString>1</gco:CharacterString>

</gmd:edition>

<gmd:editionDate>

<gco:Date>2007-08-01</gco:Date>

</gmd:editionDate>

<gmd:series>

<gmd:CI_Series>

<gmd:name>

<gco:CharacterString>Série Ortofotocartográfica IGP</gco:CharacterString>

</gmd:name>

</gmd:CI_Series>

</gmd:series>

</gmd:CI_Citation>

</gmd:citation>

<gmd:abstract>

<gco:CharacterString>Folha da série ortofotocartográfica digital do território continental, com resolução de 0.5 m, a quatro cores (RGB+IV), obtida por mosaico de fotografia aérea orto-rectificada. O voo foi efectuado com a câmara fotogramétrica digital DMC. A série compreende uma divisão em 4790 ficheiros de 4 km x 5 km nas direcções E-O e N-S, respectivamente. Possui um período de renovação bianual. Abrange zonas das freguesias de SARILHOS PEQUENOS, MOITA, MONTIJO, SARILHOS GRANDES, AFONSOEIRO, SAMOUCO e GAIO-ROSARIO, concelhos de MOITA, ALCOCHETE e MONTIJO.</gco:CharacterString>

</gmd:abstract>

<gmd:purpose>

<gco:CharacterString>Destacam-se o suporte a sistemas de informação e actualização de cartografia vectorial à escala 1:10.000.</gco:CharacterString>

</gmd:purpose>

<gmd:credit>

<gco:CharacterString> Instituto Geográfico Português</gco:CharacterString>

</gmd:credit>

```
                <gmd:pointOfContact>
                    <gmd:CI_ResponsibleParty>
                        <gmd:individualName>
                            <gco:CharacterString>DIRECÇÃO DE SERVIÇOS DE GEODESIA E
CARTOGRAFIA (DSGC)</gco:CharacterString>
                        </gmd:individualName>
                        <gmd:organisationName>
                            <gco:CharacterString>INSTITUTO                    GEOGRÁFICO
PORTUGUÊS</gco:CharacterString>
                        </gmd:organisationName>
                        <gmd:contactInfo>
                            <gmd:CI_Contact>
                                <gmd:phone>
                                    <gmd:CI_Telephone>
                                        <gmd:voice>

<gco:CharacterString>+351213819600</gco:CharacterString>
                                        </gmd:voice>
                                        <gmd:facsimile>

<gco:CharacterString>+351213819699</gco:CharacterString>
                                        </gmd:facsimile>
                                    </gmd:CI_Telephone>
                                </gmd:phone>
                                <gmd:address>
                                    <gmd:CI_Address>
                                        <gmd:deliveryPoint>
                                            <gco:CharacterString>RUA   ARTILHARIA   UM,
107</gco:CharacterString>
                                        </gmd:deliveryPoint>
                                        <gmd:city>

<gco:CharacterString>LISBOA</gco:CharacterString>
                                        </gmd:city>
                                        <gmd:postalCode>
                                            <gco:CharacterString>1099-052
LISBOA</gco:CharacterString>
                                        </gmd:postalCode>
```

<gmd:country>

<gco:CharacterString>PORTUGAL</gco:CharacterString>

</gmd:country>

<gmd:electronicMailAddress>

<gco:CharacterString>loja@igeo.pt</gco:CharacterString>

</gmd:electronicMailAddress>

</gmd:CI_Address>

</gmd:address>

</gmd:CI_Contact>

</gmd:contactInfo>

<gmd:role>

<gmd:CI_RoleCode codeList="#CI_RoleCode" codeListValue="pointOfContact">contacto</gmd:CI_RoleCode>

</gmd:role>

</gmd:CI_ResponsibleParty>

</gmd:pointOfContact>

<gmd:descriptiveKeywords>

<gmd:MD_Keywords>

<gmd:keyword>

<gco:CharacterString>Imagem</gco:CharacterString>

</gmd:keyword>

<gmd:keyword>

<gco:CharacterString>SIG</gco:CharacterString>

</gmd:keyword>

<gmd:keyword>

<gco:CharacterString>Orto-rectificação</gco:CharacterString>

</gmd:keyword>

<gmd:keyword>

<gco:CharacterString>Fotografia Aérea Digital</gco:CharacterString>

</gmd:keyword>

<gmd:type>

<gmd:MD_KeywordTypeCode codeList="#MD_KeywordTypeCode" codeListValue="discipline">disciplinar</gmd:MD_KeywordTypeCode>

</gmd:type>

```
                    </gmd:MD_Keywords>
                </gmd:descriptiveKeywords>
                <gmd:resourceConstraints>
                    <gmd:MD_LegalConstraints>
                        <gmd:useLimitation>
                            <gco:CharacterString></gco:CharacterString>
                        </gmd:useLimitation>
                        <gmd:accessConstraints>
                            <gmd:MD_RestrictionCode       codeList="#MD_RestrictionCode"
codeListValue="copyright">direitosDeAutor</gmd:MD_RestrictionCode>
                        </gmd:accessConstraints>
                        <gmd:accessConstraints>
                            <gmd:MD_RestrictionCode       codeList="#MD_RestrictionCode"
codeListValue="license">sujeitoALicenciamento</gmd:MD_RestrictionCode>
                        </gmd:accessConstraints>
                        <gmd:useConstraints>
                            <gmd:MD_RestrictionCode       codeList="#MD_RestrictionCode"
codeListValue="copyright">direitosDeAutor</gmd:MD_RestrictionCode>
                        </gmd:useConstraints>
                        <gmd:useConstraints>
                            <gmd:MD_RestrictionCode       codeList="#MD_RestrictionCode"
codeListValue="license">sujeitoALicenciamento</gmd:MD_RestrictionCode>
                        </gmd:useConstraints>
                    </gmd:MD_LegalConstraints>
                </gmd:resourceConstraints>
                <gmd:spatialRepresentationType>
                    <gmd:MD_SpatialRepresentationTypeCode
codeList="#MD_SpatialRepresentationTypeCode"
codeListValue="grid">matricial</gmd:MD_SpatialRepresentationTypeCode>
                </gmd:spatialRepresentationType>
                <gmd:spatialResolution>
                    <gmd:MD_Resolution>
                        <gmd:distance>
                            <gco:Distance uom=" meters">0.5</gco:Distance>
                        </gmd:distance>
                    </gmd:MD_Resolution>
                </gmd:spatialResolution>
                <gmd:language>
```

```
                    <gmd:LanguageCode    codeList="    LanguageCode"    codeListValue="
por">por</gmd:LanguageCode>
                </gmd:language>
                <gmd:topicCategory>

<gmd:MD_TopicCategoryCode>imageryBaseMapsEarthCover</gmd:MD_TopicCategoryCode>
                </gmd:topicCategory>
                <gmd:extent>
                    <gmd:EX_Extent>
                        <gmd:description>
                            <gco:CharacterString></gco:CharacterString>
                        </gmd:description>
                        <gmd:geographicElement>
                            <gmd:EX_GeographicBoundingBox>
                                <gmd:extentTypeCode>
                                    <gco:Boolean>1</gco:Boolean>
                                </gmd:extentTypeCode>
                                <gmd:westBoundLongitude>
                                    <gco:Decimal>-9.006988111000</gco:Decimal>
                                </gmd:westBoundLongitude>
                                <gmd:eastBoundLongitude>
                                    <gco:Decimal>-8.960479988889</gco:Decimal>
                                </gmd:eastBoundLongitude>
                                <gmd:southBoundLatitude>
                                    <gco:Decimal>38.674508566667</gco:Decimal>
                                </gmd:southBoundLatitude>
                                <gmd:northBoundLatitude>
                                    <gco:Decimal>38.719210260000</gco:Decimal>
                                </gmd:northBoundLatitude>
                            </gmd:EX_GeographicBoundingBox>
                        </gmd:geographicElement>
                    </gmd:EX_Extent>
                </gmd:extent>
                <gmd:extent>
                    <gmd:EX_Extent>
                        <gmd:temporalElement>
```

```
                            <gmd:EX_TemporalExtent>
                                <gmd:extent>
                                    <gml:TimePeriod gml:id="foo">
                                        <gml:beginPosition>2008-01-
01</gml:beginPosition>
                                        <gml:endPosition>2008-12-31</gml:endPosition>
                                    </gml:TimePeriod>
                                </gmd:extent>
                            </gmd:EX_TemporalExtent>
                        </gmd:temporalElement>
                    </gmd:EX_Extent>
                </gmd:extent>
            </gmd:MD_DataIdentification>
        </gmd:identificationInfo>
        <gmd:distributionInfo>
            <gmd:MD_Distribution>
                <gmd:distributionFormat>
                    <gmd:MD_Format>
                        <gmd:name>
                            <gco:CharacterString>TIFF + World File</gco:CharacterString>
                        </gmd:name>
                        <gmd:version>
                            <gco:CharacterString>TIFF 6.0</gco:CharacterString>
                        </gmd:version>
                    </gmd:MD_Format>
                </gmd:distributionFormat>
                <gmd:distributionFormat>
                    <gmd:MD_Format>
                        <gmd:name>
                            <gco:CharacterString>ERMAPPER ECW</gco:CharacterString>
                        </gmd:name>
                        <gmd:version>
                            <gco:CharacterString>2.0</gco:CharacterString>
                        </gmd:version>
                    </gmd:MD_Format>
                </gmd:distributionFormat>
```

```
                <gmd:transferOptions>
                    <gmd:MD_DigitalTransferOptions>
                        <gmd:unitsOfDistribution>
                            <gco:CharacterString>Seccionamento de 4 km x 5
km</gco:CharacterString>
                        </gmd:unitsOfDistribution>
                        <gmd:transferSize>
                            <gco:Real>320</gco:Real>
                        </gmd:transferSize>
                        <gmd:onLine>
                            <gmd:CI_OnlineResource>
                                <gmd:linkage>
                                    <gmd:URL>http://www.igeo.pt</gmd:URL>
                                </gmd:linkage>
                                <gmd:function>
                                    <gmd:CI_OnLineFunctionCode
codeList="#CI_OnLineFunctionCode"
codeListValue="information">informação</gmd:CI_OnLineFunctionCode>
                                </gmd:function>
                            </gmd:CI_OnlineResource>
                        </gmd:onLine>
                    </gmd:MD_DigitalTransferOptions>
                </gmd:transferOptions>
            </gmd:MD_Distribution>
        </gmd:distributionInfo>
        <gmd:dataQualityInfo>
            <gmd:DQ_DataQuality>
                <gmd:scope>
                    <gmd:DQ_Scope>
                        <gmd:level>
                            <gmd:MD_ScopeCode                    codeList="#MD_ScopeCode"
codeListValue="dataset">dataset</gmd:MD_ScopeCode>
                        </gmd:level>
                        <gmd:levelDescription>
                            <gmd:MD_ScopeDescription>
                                <gmd:dataset>
                                    <gco:CharacterString>004323B</gco:CharacterString>
```

```
                                </gmd:dataset>
                            </gmd:MD_ScopeDescription>
                        </gmd:levelDescription>
                    </gmd:DQ_Scope>
                </gmd:scope>
                <gmd:lineage>
                    <gmd:LI_Lineage>
                        <gmd:statement>
                            <gco:CharacterString>Imagem resultante do mosaico de fotografia aérea digital orto-rectificada obtida em 2007.</gco:CharacterString>
                        </gmd:statement>
                    </gmd:LI_Lineage>
                </gmd:lineage>
            </gmd:DQ_DataQuality>
        </gmd:dataQualityInfo>
        <Esri>
            <resourceType>004</resourceType>
        </Esri>
    </gmd:MD_Metadata>
```

ANEXO II – Código de Transformação dos Formatos de Dados de Entrada entre os Programas RuleGen e Weka

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.ArrayList;

/**
 *
 */

/**
 * @author grevez
 *
 *         <pre>
 * A classe RuleGen2Weka transforma os ficheiros da amostra de dados gerados
pelo
 * programa RuleGen (plugin do programa Envi para executar arvores de decisão
com
 * recurso a vários algoritmos) no formato de dados interpretavel pelo
programa Weka
 * ( o Weka é um programa desenvolvido em Java pela Universidade Waikato e que
implementa
 * vários algoritmos estatisticos e de árvores de decisão). De seguida
apresentam-se
 * exemplos dos vários formatos dos ficheiros.
 * </pre>
 *
 *         <pre>
 *
 * INPUT FILE example (jacarta_data.txt):
 *
 * "regadio",152.000,79.0000,112.000,112.000,164.000,297
 *
 * "regadio",158.000,77.0000,109.000,108.000,161.000,297
 *
 * "regadio",145.000,80.0000,113.000,113.000,162.000,297
 *
 * "regadio",143.000,78.0000,111.000,110.000,157.000,297
 *
 * "soloadescoberto",72.0000,178.000,168.000,147.000,170.000,9
 *
 * "soloadescoberto",70.0000,177.000,167.000,146.000,168.000,9
 *
 * "soloadescoberto",76.0000,176.000,167.000,146.000,171.000,175
 *
 * "floresta",200.000,76.0000,102.000,92.0000,156.000,337
 *
 * "floresta",191.000,79.0000,104.000,98.0000,162.000,337
 *
 * "floresta",177.000,80.0000,104.000,86.0000,136.000,764
 *
 * ....
 * ....
 * </pre>
 *
```

```
 *         <pre>
 *     The order of the attribute are:
 *     B1 = ndvi
 *     B2 = B1
 *     B3 = B2
 *     B4 = B3
 *     B5 = B4
 *     B6 = seg
 * </pre>
 *
 *         <pre>
 * Example Ouput file generated to read from Weka program
 *
 * @relation jecarta
 *
 * @attribute class {"regadio", "soloadescoberto", "floresta", "casas",
"estradas", "mato"}
 * @attribute B1 real
 * @attribute B2 real
 * @attribute B3 real
 * @attribute B4 real
 * @attribute B5 real
 * @attribute B6 real
 *
 * @data
 *
 * "regadio",152.000,79.0000,112.000,112.000,164.000,297
 * "regadio",158.000,77.0000,109.000,108.000,161.000,297
 * "regadio",145.000,80.0000,113.000,113.000,162.000,297
 * "regadio",143.000,78.0000,111.000,110.000,157.000,297
 * "soloadescoberto",72.0000,178.000,168.000,147.000,170.000,9
 *
 * ...
 * ...
 * </pre>
 *
 *
 */
public class RuleGen2Weka {

	private ArrayList<String> buffer;
	private ArrayList<String> classList;
	private static final String FILE_INPUT_NAME = "jecarta_data.txt";
	private static final String FILE_OUTPUT_NAME = "jecarta.arff";

	/**
	 *
	 */
	public RuleGen2Weka() {
		setBuffer(new ArrayList<String>());
		setClassList(new ArrayList<String>());
	}

	/**
	 * @param fileName
	 * @return
	 */
	private InputStream readFile(String fileName) {
		FileInputStream fio = null;
		try {
			fio = new FileInputStream(new File(fileName));
		} catch (FileNotFoundException e) {
			e.printStackTrace();
		}

		return fio;
	}
```

```
	/**
	 *
	 */
	private void writeWekaInputFile() {
		FileOutputStream fout = null;
		try {
			fout = new FileOutputStream(FILE_OUTPUT_NAME);
		} catch (FileNotFoundException e) {
			e.printStackTrace();
		}
		BufferedWriter bufferWriter = new BufferedWriter(new
OutputStreamWriter(fout));

		String relation = "@relation jecarta";
		String attribute = "@attribute";
		String data = "@data";

		try {
			bufferWriter.write(relation);
			bufferWriter.write("\n\n");
			bufferWriter.write(attribute + " class {");
			for (int i = 0; i < classList.size(); i++) {
				bufferWriter.write(classList.get(i));
				if (i < classList.size() - 1)
					bufferWriter.write(",");
			}
			bufferWriter.write("}");
			bufferWriter.write("\n");
			bufferWriter.write(attribute + " B1 real\n");
			bufferWriter.write(attribute + " B2 real\n");
			bufferWriter.write(attribute + " B3 real\n");
			bufferWriter.write(attribute + " B4 real\n");
			bufferWriter.write(attribute + " B5 real\n");
			bufferWriter.write(attribute + " B6 real\n");
			bufferWriter.write("\n\n\n");
			bufferWriter.write(data + "\n\n");
			for (int i = 0; i < buffer.size(); i++) {
				bufferWriter.write(buffer.get(i));
				bufferWriter.write("\n");
			}

			bufferWriter.flush();
			fout.flush();
			bufferWriter.close();
			fout.close();
		} catch (IOException e) {
			e.printStackTrace();
		}
		return;
	}

	/**
	 *
	 */
	private void readInputFileFromRuleGen() {
		FileInputStream fio = (FileInputStream)
this.readFile(FILE_INPUT_NAME);
		BufferedReader bufferReader = new BufferedReader(new
InputStreamReader(fio));
		String line;
		try {
			line = bufferReader.readLine();
			while (line != null) {
				// append input data to buffer, ignore blank lines
				if (line.length() > 0) {
					buffer.add(line);
					// extract class data
					extractClassNameFromInputFile(line);
```

```
                        }
                        line = bufferReader.readLine();
                }
        } catch (IOException e) {
                e.printStackTrace();
        }
}

/**
 * @param line
 */
private void extractClassNameFromInputFile(String line) {
        String className = line.substring(0, line.indexOf(","));
        if (!classList.contains(className)) {
                classList.add(className);
        }
}

/*
 * (non-Javadoc)
 *
 * @see java.lang.Object#toString()
 */
@Override
public String toString() {
        StringBuffer output = new StringBuffer();
        output.append("Input File Data:");
        output.append(buffer.toString());
        output.append("classList:");
        for (int i = 0; i < classList.size(); i++) {
                output.append(classList.get(i));
        }
        return output.toString();
}

/**
 * @param args
 */
public static void main(String[] args) {
        RuleGen2Weka ruleGen2WekaClass = new RuleGen2Weka();
        System.out.println("reading input file...");
        ruleGen2WekaClass.readInputFileFromRuleGen();
        System.out.println(ruleGen2WekaClass.toString());
        System.out.println("writing the outputfile...");
        ruleGen2WekaClass.writeWekaInputFile();
}

/**
 * @return the buffer
 */
public ArrayList<String> getBuffer() {
        return buffer;
}

/**
 * @param buffer
 *          the buffer to set
 */
public void setBuffer(ArrayList<String> buffer) {
        this.buffer = buffer;
}

/**
 * @return the classList
 */
public ArrayList<String> getClassList() {
        return classList;
}
```

```
	/**
	 * @param classList
	 *            the classList to set
	 */
	public void setClassList(ArrayList<String> classList) {
		this.classList = classList;
	}

}
```

ANEXO III – Código de Construção do formato de Arvores de Decisão do Programa Envi através do resultado obtido do programa Weka com o algoritmo C4.5

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Stack;

/**
 * @author grevez
 *
 *         <pre>
 * A classe Weka2EnviDT constroi uma árvore de decisão no formato das árvores
de decisão do
 * ENVI a partir dos resultados obtidos na geração de árvores de decisão do
programa Weka
 * </pre>
 *
 *         INPUT file example (weka_results.txt):
 *
 *         B3 <= 116 | B6 <= 297: regadio (4.0) | B6 > 297 | | B1 <= 169: mato
 *         (4.0) | | B1 > 169: floresta (6.0) B3 > 116 | B4 <= 131: casas
(4.0)
 *         | B4 > 131 | | B1 <= 17: estradas (3.0) | | B1 > 17:
soloadescoberto
 *         (3.0)
 *
 *
 *         OUTPUT file example (decision_tree.txt):
 *
 *         ENVI Decision Tree Text File (version=1.0)
 *
 *         begin node name = "B3 le 116" type = Decision location = 1,1
 *         expression = "B3 le 116" end node
 *
 *         begin node name = "B6 le 297" type = Decision location = 2,2 parent
 *         name = "B3 le 116" parent decision = true expression = "B6 le 297"
 *         end node
 *
 */
public class Weka2EnviDT {

       private static final String FILE_INPUT_NAME = "weka_results.txt";
       private static final String FILE_OUTPUT_NAME = "decision_tree.txt";

       private ArrayList<String> variables = null;
       private static final Map<String, String> rgbValues = new HashMap<String,
String>() {
              /**
               *
               */
```

```
		private static final long serialVersionUID =
5376716911884409333L;
		{
			put("estradas", "255,255,255");
			put("casas", "255,0,0");
			put("floresta", "0,255,0");
			put("regadio", "46,139,87");
			put("mato", "255,255,0");
			put("soloadescoberto", "176,48,96");
			put("aquicultura-lodo", "0,255,255");
		}
	};

	/**
	 * @author grevez
	 *
	 */
	public class EnviNode {
		private String name;
		private String type;
		private String location;
		private String expression;
		private boolean parentDecision;
		private String parentName;
		private String classValue;
		private String classRgb;
		private int pos;

		/**
		 *
		 */
		public EnviNode() {
		}

		/**
		 * @param name
		 * @param type
		 * @param location
		 * @param expression
		 * @param parentDecision
		 * @param parentName
		 * @param classValue
		 * @param classRgb
		 * @param pos
		 */
		public EnviNode(String name, String type, String location, String
expression, boolean parentDecision,
				String parentName, String classValue, String
classRgb, int pos) {
			super();
			this.name = name;
			this.type = type;
			this.location = location;
			this.expression = expression;
			this.parentDecision = parentDecision;
			this.parentName = parentName;
			this.classValue = classValue;
			this.classRgb = classRgb;
			this.pos = pos;
		}

		/**
		 * @param name
		 * @param type
		 * @param location
		 * @param expression
		 * @param pos
		 */
```

```
		public EnviNode(String name, String type, String location, String
expression, int pos) {
			this.name = name;
			this.type = type;
			this.location = location;
			this.expression = expression;
			this.pos = pos;
		}

		/**
		 * @return
		 */
		public String createRootNode() {
			StringBuffer node = new StringBuffer();
			node.append("begin node\n");
			node.append("\tname = \"" + name + "\"\n");
			node.append("\ttype = " + type + "\n");
			node.append("\tlocation = " + location + "\n");
			node.append("\texpression = \"" + expression + "\"\n");
			node.append("end node\n\n");
			return node.toString();
		}

		/**
		 * @return
		 */
		public String createNode() {
			StringBuffer node = new StringBuffer();
			node.append("begin node\n");
			node.append("\tname = \"" + name + "\"\n");
			node.append("\ttype = " + type + "\n");
			node.append("\tlocation = " + location + "\n");
			node.append("\tparent name = \"" + parentName + "\"\n");
			String decision = (parentDecision) ? "true" : "false";
			node.append("\tparent decision = " + decision + "\n");
			if (type.equalsIgnoreCase("Result")) {
				node.append("\tclass value = " + classValue + "\n");
				classRgb = (rgbValues.containsKey(name)) ?
rgbValues.get(name) : "";
				node.append("\tclass rgb = " + classRgb + "\n");
			} else {
				node.append("\texpression = \"" + expression +
"\"\n");
			}
			node.append("end node\n\n");
			return node.toString();
		}

	}

	/**
	 *
	 */
	public Weka2EnviDT() {
		variables = new ArrayList<String>();
	}

	/**
	 * @param fileName
	 * @return
	 */
	private InputStream readFile(String fileName) {
		FileInputStream fio = null;
		try {
			fio = new FileInputStream(new File(fileName));
		} catch (FileNotFoundException e) {
			e.printStackTrace();
		}
```

```
			return fio;
		}

		/**
		 * @param line
		 * @return
		 */
		private EnviNode parseRootNode(String line) {
			String[] splitLine = line.split(" ");
			String var1 = splitLine[0];
			String operator = splitLine[1];
			String value = splitLine[2];
			String name = var1 + " " + getStringOperator(operator) + " " +
value;
			EnviNode rootNode = new EnviNode(name, "Decision", "1,1", name,
1);
			return rootNode;
		}

		/**
		 * @param operator
		 * @return
		 */
		private String getStringOperator(String operator) {
			if (operator.equalsIgnoreCase("<="))
				return "le";
			else if (operator.equalsIgnoreCase("<"))
				return "lt";
			else if (operator.equalsIgnoreCase("=>"))
				return "ge";
			else if (operator.equalsIgnoreCase(">"))
				return "gt";
			else if (operator.equalsIgnoreCase("="))
				return "eq";
			return "";
		}

		/**
		 *
		 */
		private void readInputFileFromWekaResults() {
			FileInputStream fio = (FileInputStream)
this.readFile(FILE_INPUT_NAME);
			BufferedReader bufferReader = new BufferedReader(new
InputStreamReader(fio));

			FileOutputStream fout = null;
			try {
				fout = new FileOutputStream(FILE_OUTPUT_NAME);
				BufferedWriter bufferWriter = new BufferedWriter(new
OutputStreamWriter(fout));
				String header = "ENVI Decision Tree Text File
(version=1.0)\n\n";
				bufferWriter.write(header);

				buildDT(bufferReader, bufferWriter);

				int pos = 1;
				for (String var : variables) {
					bufferWriter.write("begin variable\n");
					bufferWriter.write("\tvariable name = \"" + var +
"\"\n");
					bufferWriter.write("\tfile name = \"" + "\"\n");
					bufferWriter.write("\tfile pos = " + (pos++) +
"\n");
					bufferWriter.write("end variable\n\n\n\n");
				}
```

```
                        bufferWriter.flush();
                        fout.flush();
                        bufferWriter.close();
                        fout.close();
                } catch (FileNotFoundException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }


        private void buildDT(BufferedReader bufferReader, BufferedWriter
bufferWriter) {
                String line;
                ArrayList<String> results = new ArrayList<String>();
                try {
                        line = bufferReader.readLine();

                        // root
                        EnviNode rootNode = null;
                        EnviNode parentNode = null;

                        boolean hasParent = false;
                        boolean isParentDecision = true;
                        boolean isRootRight = false;
                        Stack<EnviNode> stackNodes = new Stack<EnviNode>();

                        while (line != null) {

                                if (!line.startsWith("|") && !isRootRight) {
                                        rootNode = parseRootNode(line);
                                        String rootNodeStr =
rootNode.createRootNode();
                                        bufferWriter.write(rootNodeStr);
                                        parentNode = rootNode;
                                        isRootRight = true;
                                        stackNodes.push(rootNode);
                                } else if (!line.startsWith("|") && isRootRight) {
                                        parentNode = rootNode;
                                        isParentDecision = false;
                                } else {

                                        // count level
                                        int level = getTreeLevel(line);

                                        // remove level delimitator
                                        for (int i = 0; i < level - 1; i++) {
                                                line = line.replaceFirst("\\|   ", "");
                                        }

                                        hasParent = hasParentBefore(line,
stackNodes.peek(), hasParent);


                                        // no parent, create
                                        if (!hasParent) {
                                                // create decision node
                                                EnviNode newParentNode =
createDecisionNode(bufferWriter, line, parentNode, isParentDecision, level);
                                                parentNode = newParentNode;
                                                if (!isParentDecision) {
                                                        isParentDecision = true;
                                                }
                                                stackNodes.push(newParentNode);

                                                if (line.indexOf(":") != -1) {
```

```
                                                // create result node
                                                createResultNode(bufferWriter,
line, results, parentNode, isParentDecision, level);

                                                if (isParentDecision) {
                                                      isParentDecision = false;
                                                }
                                                hasParent = true;
                                          }
                                    }

                                    // has parent
                                    else {
                                          parentNode = stackNodes.pop();
                                          // has result
                                          if (line.indexOf(":") != -1) {
                                                createResultNode(bufferWriter,
line, results, parentNode, isParentDecision, level);
                                          }
                                          hasParent = false;
                                          isParentDecision = false;
                                    }
                              }

                              bufferWriter.flush();
                              line = bufferReader.readLine();

                        }
                  } catch (IOException e) {
                        e.printStackTrace();
                  }
            }

            /**
             * @param line
             * @param parentNode
             * @param hasParent
             * @return
             */
            private boolean hasParentBefore(String line, EnviNode parentNode,
boolean hasParent) {
                  // check if hasParent from previous stack nodes
                  String condition = (line.indexOf(":")!=-1)?line.substring(0,
line.indexOf(":")):line;
                  String[] splitLine = condition.split(" ");
                  String var1 = splitLine[0];
                  String value = splitLine[2];

                  String[] splitLineParent = parentNode.expression.split(" ");
                  String var1Parent = splitLineParent[0];
                  String valueParent = splitLineParent[2];

                  if (var1.equalsIgnoreCase(var1Parent) &&
value.equalsIgnoreCase(valueParent)) {
                        hasParent = true;
                  }
                  return hasParent;
            }

            /**
             * @param bufferWriter
             * @param line
             * @param results
             * @param parentNode
             * @param isParentDecision
             * @param level
             * @throws IOException
             */
```

```
	private void createResultNode(BufferedWriter bufferWriter, String line,
ArrayList<String> results,
			EnviNode parentNode, boolean isParentDecision, int level)
throws IOException {
		String nameResult = line.substring(line.indexOf(":") + 2,
line.indexOf("(") - 1);

		int newPos = (isParentDecision) ? (parentNode.pos * 2) :
(parentNode.pos * 2) - 1;

		if (!results.contains(nameResult))
			results.add(nameResult);

		int posClass = results.indexOf(nameResult);

		String locationResultStr = (level + 1) + "," + newPos;
		EnviNode nodeResult = new EnviNode(nameResult, "Result",
locationResultStr, nameResult, isParentDecision,
				parentNode.name, Integer.toString(posClass + 1),
null, newPos);
		String nodeResultStr = nodeResult.createNode();
		//System.out.println(nodeResultStr);
		bufferWriter.write(nodeResultStr);
	}



	/**
	 * @param bufferWriter
	 * @param line
	 * @param parentNode
	 * @param notParentDecision
	 * @param level
	 * @return
	 * @throws IOException
	 */
	private EnviNode createDecisionNode(BufferedWriter bufferWriter, String
line, EnviNode parentNode,
			boolean isParentDecision, int level) throws IOException {
		String condition = (line.indexOf(":")!=-1)?line.substring(0,
line.indexOf(":")):line;
		String[] splitLine = condition.split(" ");
		String var1 = splitLine[0];
		if (!variables.contains(var1))
			variables.add(var1);
		String operator = splitLine[1];
		String value = splitLine[2];

		String name = var1 + " " + getStringOperator(operator) + " " +
value;

		int newPos = (isParentDecision) ? (parentNode.pos * 2) :
(parentNode.pos * 2) - 1;

		String locationStr = level + "," + newPos;
		EnviNode newParentNode = new EnviNode(name, "Decision",
locationStr, name, isParentDecision, parentNode.name, null,
				null, newPos);
		String nodeStr = newParentNode.createNode();
		//System.out.println(nodeStr);
		bufferWriter.write(nodeStr);
		return newParentNode;
	}

	/**
	 * @param line
	 * @return
	 */
```

```
	private int getTreeLevel(String line) {
		int level = 1; // inital level 1 set by envi
		int pos = 0;
		String delimitator = "|   ";
		for (int i = 0; i < line.length(); i++) {
			if (line.indexOf(delimitator, pos) != -1) {
				level++;
				pos = line.indexOf(delimitator, pos) +
delimitator.length();
			} else
				break;
		}
		return level;
	}

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		Weka2EnviDT weka2EnviDt = new Weka2EnviDT();
		System.out.println("writing the outputfile...");
		weka2EnviDt.readInputFileFromWekaResults();
	}

}
```